
Proceedings Paper:
Peng, P. (2020) Robust clustering oracle and local reconstructor of cluster structure of graphs. In: Chawla, S., (ed.) 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2020). 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2020), 05-08 Jan 2020, Salt Lake City, Utah, USA. SIAM Proceedings. Society for Industrial and Applied Mathematics (SIAM). ISBN: 9781611975994.

https://doi.org/10.1137/1.9781611975994.179

Robust Clustering Oracle and Local Reconstructor of Cluster Structure of Graphs

Pan Peng*

Abstract

We develop sublinear time algorithms for analyzing the cluster structure of graphs with noisy partial information. A graph G with maximum degree at most d is called $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable, if it can be partitioned into at most k parts, such that each part has inner conductance at least ϕ_{in} and outer conductance at most ϕ_{out}, where d is assumed to be constant. A graph G is called to be an ε-perturbation of a $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable graph if there is partition of G with at most k parts (called clusters), such that one can insert/delete at most εdn *intra-cluster* edges to make it a $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable graph. We are given query access to the adjacency list of such a graph.

We show that one can construct in $O(\sqrt{n} \cdot \text{poly}(\frac{k \log n}{\phi\varepsilon}))$ time a *robust clustering oracle* for a bounded-degree graph G that is an ε-perturbation of a $(k, \phi, O(\frac{\varepsilon\phi}{k^3 \log n}))$-clusterable graph. Using such an oracle, a typical clustering query (e.g., IsOutlier(s), SameCluster(s, t)) can be answered in $O(\sqrt{n} \cdot \text{poly}(\frac{k \log n}{\phi\varepsilon}))$ time and the answers are consistent with a partition of G in which all but $O(k\sqrt{\frac{\varepsilon}{\phi}}n)$ vertices belong to a good cluster, i.e., a set with inner conductance at least $\frac{\phi}{2}$, and outer conductance $O(\frac{\sqrt{\varepsilon}\phi^{1.5}}{k^4 \log n})$. We also develop a *local reconstruction* algorithm that takes as input a graph as above, and on any query vertex v, outputs all its neighbors in the reconstructed graph G', which is guaranteed to be $(k, \Omega(\frac{\varepsilon\phi}{k^4 \log n}), 1)$-clusterable (with slightly boosting degree bound). The number of edges changed is at most $O(k\sqrt{\frac{\varepsilon}{\phi}}n)$. Furthermore, the algorithm runs in $O(\sqrt{n} \cdot \text{poly}(\frac{k \log n}{\phi\varepsilon}))$ time (per query) and can answer consistently with the same G' for any sequence of queries it gets.

*Department of Computer Science, University of Sheffield, Sheffield, U.K. Email: p.peng@sheffield.ac.uk.

1 Introduction

Graph clustering is a fundamental task arising from many domains, including computer science, social science, network analysis and statistics. Given a graph, the task is to group the vertices into *reasonably good* clusters, where vertices inside the same cluster are well-connected to each other, and any two different clusters are well-separated (see e.g., surveys [Sch07, POM09, For10, New12]). Due to the massive size of modern network data, local algorithms that run in *sublinear* time for analyzing the cluster structure of the graph are receiving growing interest. Such algorithms are typically assumed to be able to explore the input graph by performing appropriate queries, e.g., query the degree or the neighbor of any node. There have been two main frameworks for designing sublinear algorithms for graph clustering, if we use the well-motivated notion *conductance* (see below) to measure the quality of clusters. In the first one, called *local graph clustering*, the goal is to find a cluster from a specified vertex with running time that is bounded in terms of the size of the output set (and with a weak dependence on n) (see e.g., [ST13, ACL06, AP09, OT12, AOPT16, ZLM13, OZ14]). If the target cluster has small enough size, then the running time of the resulting algorithm will be sublinear in the input size. In the second one, called *testing cluster structure* in the framework of *property testing*, the goal is to distinguish if an input graph has a typical cluster structure or is far from such cases (see [CPS15, CKK$^+$18] and more discussions in Appendix A). Such algorithms make decisions on the global cluster structure of the input graph by sampling vertices and locally exploring a small portion of the graph, and they can serve as a preliminary step before learning the cluster structure.

In this work, we study local and sublinear algorithms for analyzing the cluster structure of graphs that may contain noise and/or outliers. In many real applications, due to external noise or errors, the network data set may fail to have the desired property (here, the cluster structure), while it might still be close to have this property. That is, the graph G under our consideration is some kind of *perturbation of a clusterable graph* or a *noisy clusterable graph*: G is first chosen from some class of clusterable graphs with an underlying while unknown partition, and then some noise and/or outliers are introduced by some adversary or in some random way. This is a relaxation of a common assumption for many existing clustering algorithms that the input graph is simply well clusterable. We would like to very efficiently process such a noisy clusterable graph and extract useful information regarding its cluster structure. Slightly more precisely, we study two types of sublinear algorithms for analyzing the cluster structure of graphs with noisy partial information.

The first type of algorithm is driven by the following natural question: *Given a noisy clusterable graph, can we build an oracle (or implicit representation) in sublinear time, that can support typical queries regarding the cluster structure of the graph in sublinear time?* For example, we would like to query "Is a vertex s a noise/outlier?". If the answer is "No", we would further like to know "Which cluster does s belong to?", and "Do s and t belong to the same cluster?", given that both vertices s, t are not outliers. We would require that all the query answers will be consistent, e.g., if u, v are reported to belong to the same cluster, v, w are reported to belong to the same cluster, then u, w will also be reported to belong to the same cluster. Furthermore, we would like to minimize the number of vertices for which the oracle returns the "wrong" answers in the sense that the output partition of the algorithm should be close to an underlying maximal good clustering of the graph. We will call such an oracle a *robust clustering oracle*. Such oracles might be already interesting from real-world applications. For example, quickly identifying outliers might be valuable in road networks and medical data. Sometimes, we only want the cluster information of a small group of vertices while do not care about other parts of the graph. Furthermore, it will be desirable to work on-the-fly on a clean data after removing a small fraction of outliers. Besides these real-world applications, such oracles might be given as input for other clustering algorithms that are equipped with the power of making the above mentioned clustering queries (see e.g., [MS17b, MS17a, AKBD16, ABJK18, ABJ18]).

Our second type of algorithm is motivated by a very related question: *Given a noisy clusterable graph, can we fix it by minimally modifying the original graph, and provide query access to the reconstructed clusterable graph in sublinear time?* We address this question in the *online reconstruction* framework

introduced by [ACCL08]. In this framework (for graphs), given a property Π and query access to a graph G that is close to have Π, we want to output a graph G' such that G' has the property Π and G is modified minimally to get G'. Furthermore, we would like to output G' in a local and consistent way that can provide query access to G' by making as few queries as possible to the input graph G. The corresponding algorithm will be called a *local reconstructor* or *local filter* for property Π [ACCL08, SS10, AT10]. The natural application of such local reconstructors is when only a small portion of the corrected graph G' is needed or when we want to make use of the graph G' in a distributed manner. (Note that in many applications, queries are made to a large graph which is assumed to exhibit some structural property.) Here, we focus on designing a local filter for cluster structure of graphs and providing consistent query access to a clusterable graph. In practice, such algorithms might be used for fast recommending products to users even if there is some noise in the data.

In this work, we give both sublinear robust clustering oracle and local reconstructors for the cluster structure of graphs. Now we give basic definitions of clusters and (noisy) clusterable graphs, formalize our algorithmic problems, state our main results and sketch our technical ideas.

1.1 Basic Definitions

Conductance based clustering. Following a recent line of research on graph clustering (e.g., [OT14, CPS15, PSZ17, DPRS19], which was built upon [KVV04]), we will use *conductance* based definition for measuring the quality of clusters and the cluster structure of graphs. In this paper, we will focus on undirected graphs with bounded maximum degree. We call an undirected graph $G = (V, E)$ a *d-bounded* graph if its maximum degree is upper bounded by some parameter d, which is always assumed to be some sufficiently large constant (at least 10). For any two subsets $S, T \subseteq V$, we let $E(S, T)$ denote the set of edges with one endpoint in S and the other point in T. The *conductance* $\phi_G(S)$ of a set S in G is defined to be the ratio between the number of edges crossing S and its complement $V \setminus S$ and the maximum number of edges possible incident to S, that is, $\phi_G(S) := \frac{|E(S,V\setminus S)|}{d|S|}$. The *conductance* $\phi(G)$ of the graph G is defined to be the minimum value of the conductance of any set S with size at most $n/2$, that is, $\phi(G) := \min_{S:|S|\leq n/2} \phi_G(S)$. For convenience, for the singleton graph G (that consists of a single vertex with no edges) we define its inner conductance $\phi(G)$ to be 1.

Given a vertex set $S \subset V$, we let $G[S]$ denote the subgraph induced by vertices in S. In the following, we will refer to $\phi_G(S)$ and $\phi(G[S])$ as the *outer conductance* and *inner conductance*, respectively. Given two parameters ϕ_{in} and ϕ_{out}, we call a set S a *(ϕ_{in}, ϕ_{out})-cluster* if

$$\phi_G(S) \leq \phi_{\text{out}}, \quad \phi(G[S]) \geq \phi_{\text{in}}.$$

For a good cluster S, we expect ϕ_{in} to be large and ϕ_{out} to be small. In particular, if $S = V$ and $\phi(G[V]) = \phi(G) \geq \phi_{\text{in}} \geq \phi$ for some constant ϕ, then we call the graph G a *ϕ-expander* which by itself is a good cluster and has been extensively studied in theoretical computer science (see e.g., [HLW06]). It is useful to note that $\phi_G(V) = 0$. When G is clear from the context, we omit the subscript G from $\phi_G(S)$. A *k-partition* of a graph $G = (V, E)$ is a partition of V into k subsets, $V_1, \cdots, V_k$ such that $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\cup_i V_i = V$. In particular, V itself is a 1-partition. We have the following definition of clusterable graphs that characterize graphs with typical cluster structure (see e.g., [OT14]).

Definition 1.1. *Given parameters $d, k, \phi_{in}, \phi_{out}$, we call a k-partition $P_1, \cdots, P_k$ of a d-bounded graph G a $(k, \phi_{\text{in}}, \phi_{\text{out}})$-*clustering *if for each $i \leq k$, $\phi(G[P_i]) \geq \phi_{in}$ and $\phi_G(P_i) \leq \phi_{out}$. A d-bounded graph G is called to be $(k, \phi_{\text{in}}, \phi_{\text{out}})$-*clusterable *if G has an $(h, \phi_{in}, \phi_{out})$-clustering for some $h \leq k$.*

Note that in our definition, a $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable graph may contain less than k clusters, and $(1, \phi_{\text{in}}, 0)$-clusterable graphs are equivalent to ϕ_{in}-expanders.

Clusterable graphs with modeling noise. We assume that the input graph to the algorithm is generated from the family of all $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable graphs and then modified by an adversary in some manner. We have the following definition.

Definition 1.2. *(Clusterable Graphs with Modeling Noise or Noisy Clusterable Graphs) In this model, the adversary first chooses an arbitrary graph G^* from the family of all $(k, \phi_{in}, \phi_{out})$-clusterable graphs with maximum degree upper bounded by d. Then the adversary may do the following:*

1. *Choose an arbitrary $(h, \phi_{in}, \phi_{out})$-clustering $P_1, \cdots, P_h$ of G^* for some $h \leq k$.*
2. *Insert and/or delete at most $\varepsilon \cdot dn$ edges (*noise*) within the clusters $G^*[P_i]$, $1 \leq i \leq h$, while preserving the degree bound.*

*We call the resulting graph G an ε-*perturbation *of G^* with respect to the h-partition $P_1, \cdots, P_h$.*

Equivalently, a graph G is called to be an ε-perturbation of a $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable graph if there is partition of G with at most k parts (called clusters), such that one can insert/delete at most εdn *intra-cluster* edges to make it a $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable graph. For simplicity, in the above definition, we only allowed the adversary to perturb the edges *inside* the clusters, while our algorithm can actually be extended to work for the case that the adversary is also allowed to perturb *inter-cluster* edges, up to a very *limited extent*[1]. This definition generalizes the notion of noisy expander graphs studied by Kale, Peres, and Seshadhri [KPS13], which correspond to $k = 1$ in our problem. In their setting, the adversary first chooses a ϕ-expander and then modifies it by inserting/deleting ε fraction of edges in the graph.

1.2 Problem Formalizations and Main Results

Now we formalize our algorithmic problems and present our main results. For a d-bounded graph G, we will assume the algorithm is given query access to the adjacency list of G, that is, in constant time we can query the i-th neighbor of any vertex v.

Robust clustering oracle. Note that in a noisy clusterable graph, if the noise is not too much, many vertices are still expected to belong to some good cluster, and those vertices that do not belong to any good cluster will intuitively correspond to *outliers* or *noise*. Given query access to the adjacency list of a d-bounded graph G that is promised to be an ε-perturbation of a $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable graph, we are interested in constructing an implicit representation, called a *robust clustering oracle*, of G in sublinear time such that typical queries regarding the cluster structure of G can be answered as quickly as possible (also in sublinear time). More precisely, the oracle should support the following types of *clustering queries*:

1) IsOutlier(s): Is a vertex s a noise/outlier?

As mentioned above, a vertex that does not belong to any good cluster should be reported as noise or outlier. For any non-outlier vertices s, t, the oracle can further support

2) WhichCluster(s): Which cluster does s belong to?

3) SameCluster(s, t): Do s and t belong to the same cluster?

In the following, without loss of generality, we will assume that for any non-outlier vertex s and the corresponding WhichCluster(s) query, the oracle will output an integer i with $1 \leq i \leq h$ that specifies the index of the cluster that s belongs to, for some integer h. Furthermore, given the ability of answering WhichCluster queries, for any two non-outlier vertices s, t, we simply define SameCluster(s, t) to be the procedure that checks if WhichCluster(s) is equal to WhichCluster(t). This will naturally ensures the consistency for SameCluster queries. Note that the output of the algorithm naturally defines a partition of V, i.e.,

$$P_i := \{u \in V : \text{WhichCluster}(u) = i\}, 1 \leq i \leq h, \quad B := \{u \in V : \text{IsOutlier}(u) = \textbf{Yes}\}.$$

We would like to minimize the number of vertices for which the oracle returns the "wrong" answers. That is, for most vertices v that do belong to some underlying good cluster in the perturbed G, we expect IsOutlier(v) to return "No". Furthermore, for most vertices u, v that belong to the same cluster (resp.

[1]More precisely, the adversary can be allowed to perturb a ϕ_{out} fraction of inter-cluster edges: this essentially can then be reduced to the case that only intra-cluster perturbations are allowed by re-scaling a constant factor of conductance values, i.e., one can view that the adversary first chooses a $(k, \phi_{\text{in}}, 2\phi_{\text{out}})$-clusterable graph and then perturbs its intra-cluster edges.

different clusters), we expect SAMECLUSTER(u, v) to return "Yes" (resp. "No"). One further crucial requirement of a robust clustering oracle and the corresponding clustering query algorithm is to maintain *consistency* among all queries. That is, on different query sequences, the answers of the oracle should be consistent with the same h-partition $D_1, \cdots, D_h$ of V for some $h \leq k$, in which all but a small fraction of vertices belong to some *good* cluster. Since the oracle construction and the corresponding query algorithm are typically randomized, we fix the randomness seed of the oracle and query algorithm once and for all to ensure consistent answers. Then the algorithm will be a *deterministic* procedure for any input query, which further guarantees that the partition $D_1, \cdots, D_h$ is determined by G and the internal randomness of the oracle and the algorithm, and is independent of the order of queries. This feature allows the oracle to be used in the distributed manner as consistency is guaranteed.

We provide the first robust clustering oracle with both sublinear preprocessing time and query time. We will assume d is a constant throughout the paper. Let $P \triangle Q$ denote the symmetric difference between two vertex sets P, Q. For two partitions $\mathcal{A} = \{A_1, \cdots, A_s\}$ and $\mathcal{A}' = \{A'_1, \cdots, A'_t\}$ of V with $s \leq t$, we define the symmetric difference between $\mathcal{A}$ and $\mathcal{A}'$ to be $|\mathcal{A} \triangle \mathcal{A}'| = \min_\sigma \sum_{i=1}^{t} |A_i \triangle A'_{\sigma(i)}|$, where $A_i = \emptyset$ for $i > s$ and σ ranges over all bijections $\sigma : \{1, \cdots, t\} \to \{1, \cdots, t\}$.

Theorem 1.3 (Robust Clustering Oracle)**.** *There exists an algorithm that takes as input parameters $n \geq 1$, $d > 10$, $k \geq 1$, $\phi \in (0, 1)$, $\varepsilon \in [\Omega(\frac{1}{n}), O(\frac{\phi}{k^2})]$ and has query access to the adjacency list of a graph $G = (V, E)$ that is an ε-perturbation of a $(k, \phi, O(\frac{\varepsilon\phi}{k^3 \log n}))$-clusterable graph, and constructs a robust clustering oracle in $O(\sqrt{n} \cdot poly(\frac{k \cdot \log n}{\phi\varepsilon}))$ pre-processing time. Furthermore, it holds that*

1. *Using the oracle, the algorithm can answer any clustering query (i.e.,* ISOUTLIER*,* WHICHCLUSTER *or* SAMECLUSTER*) in $O(\sqrt{n} \cdot poly(\frac{k \cdot \log n}{\phi\varepsilon}))$ time.*
2. *There exists a partition $\mathcal{A}' = \{D_1, \cdots, D_{h'}, B'\}$ of G, for some $h' \leq k$, such that*
   - *the partition only depends on G and the input parameters of the algorithm, and is independent of the order of queries;*
   - *it holds that each D_i is a $(\frac{\phi}{2}, O(\frac{\sqrt{\varepsilon}\phi^{1.5}}{k^3 \log n}))$-cluster, for any $1 \leq i \leq h'$; and*
   - *with probability at least $1 - \frac{1}{n}$, the partition $\mathcal{A} = \{P_1, \cdots, P_h, B\}$ output by the algorithm satisfies that $h' \leq h \leq k$ and $|\mathcal{A} \triangle \mathcal{A}'| = O(k\sqrt{\frac{\varepsilon}{\phi}}n)$.*

We remark that there is no algorithm that allows both $o(\sqrt{n})$ pre-processing time and $o(\sqrt{n})$ query time for ISOUTLIER queries, as otherwise, one could obtain a property testing algorithm for expansion with $o(\sqrt{n})$ queries, which will be a contradiction to a known lower bound [GR00] (see discussions on relation to property testing in Appendix A). Furthermore, the second item of the theorem implies that the total number of vertices that are reported as outliers is at most $O(k\sqrt{\varepsilon/\phi} \cdot n)$ and that the query answers are consistent with a partition of G in which all but $O(k\sqrt{\varepsilon/\phi} \cdot n)$ vertices belong to a $(\frac{\phi}{2}, O(\frac{\sqrt{\varepsilon}\phi^{1.5}}{k^3 \log n}))$-cluster. We also note that in the statement of the above theorem, the range of ε is[2] $\varepsilon \in [\Omega(\frac{1}{n}), O(\frac{\phi}{k^2})]$. If $\varepsilon = \Omega(\phi/k^2)$, the noise will be too much and our algorithm cannot locally identify even one cluster. Removing the $\log n$ gap between the inner conductance and outer conductance seems to be hard, at least for methods that are based on random walk distances (as we used here). For example, in [CKK+18], it has been discussed that in general, it is impossible to use Euclidean distance between random walk distributions to test 2-clusterability if one wants the gap to be a constant. (Testing 2-clusterability is easier than the robust clustering oracle problem; see Appendix A.) On the other hand, being able to correctly answer SAMECLUSTER(u, v) queries intuitively requires or induces a distance based approach, as the vertices in the same cluster are "similar" or "close to" each other, while vertices in different clusters are "dissimilar" or "far from" each other.

[2]Since $\varepsilon = O(\phi/k^2)$, we do not see the ϕ^2 dependency (from Cheeger inequality) between the outer and inner conductance.

Local reconstructor of graph cluster structure. We are interested in designing a local reconstruction algorithm for the cluster structure of graphs. Given query access to the adjacency list of a d-bounded graph G that is promised to be an ε-perturbation of a $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable graph, our goal is to design a *local filter* that provides query access to a $(k, \phi'_{\text{in}}, \phi'_{\text{out}})$-clusterable graph G' such that the distance between G and G' is as close as possible. That is, we would like to output G' in a local manner that for any vertex query, the neighborhood of v, i.e., the set of all neighbors of v, in G' can be answered in sublinear time (in particular, by making as few queries to the adjacency list to G as possible). Similar as for the robust clustering oracle, it is crucial to require a local filter to maintain *consistency* among all queries. Here we require that for different query sequences, the answers of the filter should be consistent with the same reconstructed graph G'. Again, the filter is suitable to be used in the distributed manner as consistency is guaranteed. In our local filter for clusterable graphs, we also aim to make the gap between $\phi_{\text{in}}, \phi_{\text{out}}$ and the gap between ϕ_{in} and ϕ'_{in} as small as possible. We next state our theorem regarding our local filter for clusterable graphs as follows.

Theorem 1.4 (Local Reconstructor of Cluster Structure)**.** *There exists a local reconstruction algorithm that takes as input parameters $n \geq 1$, $d > 10$, $k \geq 1$, $\phi \in (0,1)$, $\varepsilon \in [\Omega(\frac{1}{n}), 1]$ and has query access to the adjacency list of a graph $G = (V, E)$ that is an ε-perturbation of a $(k, \phi, O(\frac{\varepsilon\phi}{k^3 \log n}))$-clusterable graph, and provides query access to a graph $G' = (V, E')$ such that the following holds with probability at least $1 - 4/n$:*

1. *G' is $(k, \Omega(\frac{\varepsilon\phi}{k^4 \log n}), 1)$-clusterable, and has maximum degree at most $d + 16$.*
2. *The number of edges changed is at most $O(\min\{1, k\sqrt{\varepsilon/\phi}\} \cdot n)$.*
3. *G' is determined by G and the internal randomness of the algorithm, and is independent of the order of queries.*
4. *On each query v, the neighborhood of v in G' can be answered in $O(\sqrt{n} \cdot poly(\frac{k \cdot \log n}{\phi\varepsilon}))$ time.*

Note that by Item 1, the resulting graph can be partitioned into at most k parts, each with relatively large inner conductance (i.e., $\Omega(\frac{\varepsilon\phi}{k^4 \log n})$), with no guarantee on outer conductance (as each set trivially has outer conductance at most 1). (Such instances are exactly the object that was studied in [CKK+18] in the framework of property testing.) By sacrificing the inner conductance quality, we can also find a clustering of G' with small outer conductance. That is, we can guarantee that G' is also $(k, \Omega(\frac{\nu^k}{6^k k^4} \frac{\varepsilon\phi}{\log n}), \min\{k\nu, 1\})$-clusterable for any $\nu \in [0,1]$ (see Appendix D for details). Item 3 implies that all query answers are consistent, that is, the vertex u is output as a neighbor of v in G' if and only if v is output as a neighbor of u. From the discussion below on the connections between our local reconstruction algorithm and property testing, the running time of our filter is optimal (in terms of dependency on n) up to polylogarithmic factors.

Furthermore, our algorithm generalizes the local reconstruction algorithm for expander graphs by [KPS13], which corresponds to the special case $k = 1$ in our problem, though our approximation ratio of the number of modified edges is worse. More precisely, for $\varepsilon = \Omega(\phi)$, both our algorithm and the algorithm in [KPS13] will add $\Theta(dn)$ edges (as the noise part is too large, and thus almost all vertices will be reported as outliers and the resulting graph G' is almost the complete hybrid of the original graph G and an explicitly constructible expander G_{exp}, i.e., the neighborhood of each vertex in G' is the union of its neighborhood in G and G_{exp}; for $\varepsilon = O(\phi)$, the algorithm in [KPS13] reconstructs a graph that is an ε-perturbation of a ϕ-expander by modifying at most $O(\varepsilon n/\phi)$ edges, and the resulting graph has conductance at least $\Omega(\phi^2/\log n)$ and maximum degree also upper bounded by[3] $d + 16$, while our algorithm has to modify

[3]Note that [KPS13] claimed that the number of modified edges is at most $O(\frac{\phi}{\log n}\varepsilon n)$ and the maximum degree of the resulting graph is $d + O(\lceil \frac{d\phi^2}{\log n} \rceil)$. However, this claim is not correct (at least for d-bounded graphs with d being constant), and the number of changed edges and the maximum degree bound from their analysis should be $O(\frac{\varepsilon}{\phi} n)$ and $d + 16$, respectively [Ses19]. They obtained their claimed results by adding $t := \lceil \frac{d\phi^2}{c \log n} \rceil$ parallel edges while repairing bad vertices, from which they get that the maximum degree is $d + 16t$ and the number of added edges to the optimal distance (i.e., εdn) is $\frac{16t}{d\phi} = O(\phi/\log n)$, which is incorrect as it always holds that $t = 1$ for constant d and large enough n.

$O(\sqrt{\varepsilon/\phi}\cdot kn)$ edges. We further note that the algorithm in [KPS13] guarantees that the reconstructed graph has inner conductance at least $\Omega(\phi^2/\log n)$, while the resulting graph from our algorithm is guaranteed to have a partition with at most k parts, each with inner conductance at least $\Omega(\varepsilon\phi/(k^4\log n))$. Removing the $\log n$ factor in the inner conductance of the output graph seems to be a very challenging task, even for the case $k=1$. See Section 6 for more discussions.

Local mixing property on noisy clusterable graphs. In order to derive the above algorithmic results, we prove an interesting behavior, which we call *local mixing property*, of random walks on noisy clusterable graphs. For technical reasons, we will consider the *uniform averaging walk of t steps* on a graph G: In this walk, we choose a number $\ell\in\{0,1,2,\cdots,t-1\}$ uniformly at random, and stop the (normal) random walk after ℓ steps. We let $\mathbf{a}_v^t$ denote the probability vector for a uniform averaging walk of t steps starting at v and let $\|\mathbf{p}_1-\mathbf{p}_2\|_{\mathrm{TV}}$ denote the total variance distance between two distributions $\mathbf{p}_1,\mathbf{p}_2$. For a set A, we let $\mathcal{U}_A$ denote the uniform distribution on A. We have the following theorem.

Theorem 1.5 (Local Mixing Property of Random Walks)**.** *Let $0<\gamma,\varepsilon<1$. Let $\phi_{out}\le\frac{a_{1.5}\varepsilon\gamma^4\phi_{in}^2}{k^3\log n}$ for some sufficiently small constant $a_{1.5}>0$. Let G be a d-bounded graph with an h-partition $C_1,C_2,\cdots,C_h$ such that $\phi_G(C_i)\le\phi_{out}$ for any $1\le i\le h\le k$. For each $i\le h$, we let $D_i\subseteq C_i$ denote a large subset of vertices such that $\phi(G[D_i])\ge\phi_{in}$, and let $B_i=C_i\setminus D_i$. If $\sum_i|B_i|\le\varepsilon n$, then for any D_j with $|D_j|\ge 3\sqrt{\varepsilon}n$, there exists a subset $\widehat{D}_j\subseteq D_j$ such that $|\widehat{D}_j|\ge(1-4\sqrt{\varepsilon})|D_j|$ such that for any $s\in\widehat{D}_j$, and $t=\frac{120\log n}{\gamma\phi_{in}^2}$, it holds that*

$$\|\boldsymbol{a}_s^t-\mathcal{U}_{C_j}\|_{TV}<\gamma+\sqrt{\varepsilon}.$$

Intuitively, the set B_i corresponds to the noisy part inside each cluster C_i and we assume that the total fraction of noisy part is parametrized by ε. Then the above theorem says that the rest of the large part (i.e., clusterable part) exhibits some nice local mixing property: for most vertices s in a large cluster C (of size $\Omega(\sqrt{\varepsilon}n)$), a uniform averaging random walk (of appropriately chosen length) from s will converge quickly to the uniform distribution on C. This is a generalization of the global mixing property of noisy expander graphs in [KPS13], though their results are stated for the more general Markov chains.

1.3 Our Techniques

To design a robust clustering oracle, we first note that it is relatively easy to design a clustering oracle without noise (if the gap between ϕ_{in} and ϕ_{out} is $O(\log n)$ as we considered here). This can be done by a refined analysis of the property testing algorithm in [CPS15] that samples a small number of vertices, and then tests if the ℓ_2 norm distance between the random walk distributions from any two vertices is larger than some threshold or not. However, the analysis depends on the spectral property (e.g., a gap between λ_k and λ_{k+1}) of clusterable graphs, and cannot be easily generalized to the case that the input graph contains noise, as such spectral property is very sensitive to noise (e.g., deleting all edges incident to a constant number of vertices will break down the property).

In order to handle noisy input, we use the ℓ_1 norm distance between the corresponding random walk distributions to test if the starting two vertices belong to the same cluster or not, and we make use of the local mixing property of random walks in Theorem 1.5. In order to prove such a mixing property, we first show that it does hold for clusterable graphs *without* noise, by exploiting a spectral property that characterizes the first k eigenvectors of clusterable graphs given by [PSZ17]. To generalize the result to a noisy clusterable graph G, we view the random walks on the graph as a Markov chain and consider a new Markov chain that is induced on vertices in the clusterable part in G. (Such a new chain has also been used in [KPS13] for analyzing noisy expanders.) We show the induced Markov chain does correspond to a clusterable graph H (by overcoming the difficulty that the outer conductance of each corresponding cluster increases and might change the cluster structure too much) and thus the random walks in H satisfy the local mixing property. However, the walks on H can be very different from the random walks in the original graph G. We then

give a novel application of an old technique called *stopping rules* of Markov chains that was introduced by Lovósz and Winkler [LW97] to relate these two walks, and bound the total variance distance between two random walk distributions from a vertex in any large cluster of G and H. This allows us to show the local mixing property in the graph G. The idea of using stopping rules to show that a random walk mixes inside a *subgraph* (i.e., cluster) rather than in the whole graph might find other applications.

Given such a local mixing property of random walks in the noisy clusterable graph, we are able to design a robust clustering oracle and the corresponding clustering query algorithm with sublinear preprocessing and query time. We first note that if the noisy part is not too large (i.e., $\varepsilon = O(\phi/k^2)$), then the graph G has a non-trivial partition $D_1, \cdots, D_{h'}, B'$ with $h' \geq 1$ that only depends on the corresponding parameters (i.e., ε, ϕ, n) and G itself, and that each D_i is a good cluster with large size (containing at least $\Omega(\sqrt{\varepsilon})$ fraction of vertices), and B' has small size. Our key idea is to use random walks to learn a succinct representation H, which is a weighted graph with roughly $O(\log n)$ vertices, of the clusterable part of graph G, such that each cluster D_i in G will be mapped to a unique clique (called a *core*) in H with appropriate edge weight. Furthermore, by using the weights and the size bounds of these cliques, we can efficiently identify them from H, and use them to answer the WHICHCLUSTER queries. Slightly more precisely, in the *preprocessing* (or *learning*) phase, the algorithm samples a set S of $\Theta(\log n)$ vertices, and uses the statistics of $\tilde{O}(\sqrt{n})$ random walks from each sampled vertex to (quite accurately) estimate the so-called *reduced collision probability (rcp)* of (the random walks of appropriate length from) any two sampled vertices that was introduced in [KPS13]. We construct a weighted *similarity graph* H on the sample set S such that the weight of each edge (u, v) is our estimate of the rcp of u, v, for any $u, v \in S$. We show that if the noisy part is not too large, then, by the aforementioned local mixing property, for (most) pair of vertices $u, v \in S \cap D_i$, the rcp of u, v will be close to $1/|D_i|$. Thus, the weight of edge (u, v) in H will be set to be a number close to $1/|D_i|$, and most vertices in $S \cap D_i$ form a clique S_i in H with edge weights close to $1/|D_i|$. We further observe that S_i has relatively large size (roughly $|S| \cdot \frac{|D_i|}{n}$), as $|D_i|$ is large; and that any vertex $v \in S_i$ can only belong to exactly one such (large) clique, as otherwise, the total probability mass of random walk distribution from v will exceed 1, which can not happen. These properties allow us to efficiently identify the unique core S_i from H that corresponds to the cluster D_i by a simple greedy algorithm and further to answer membership queries. We remark that in [CPS15], a similarity graph is also constructed, while that graph is unweighted and only tells if the original graph is k-clusterable or not according to the number of connected components, which is far from sufficient for our application.

In the *query* phase, we check if the queried vertex v belongs to any of the learned cores or not to decide if it is an outlier or not. This, again, can be done by estimating the rcp of the walks from v and other vertices in S (by running $\tilde{O}(\sqrt{n})$ random walks), and is guaranteed by the local mixing property of random walks. In particular, for most vertices v in a cluster D_i, the rcp of random walks from v and any other vertex that is in S_i corresponding to D_i will be also around $1/|D_i|$. If this is the case, we output i as the index of the cluster that v belongs to; otherwise, we report it as an outlier. The above analysis shows that most vertices in $D_1, \cdots, D_{h'}$ will be correctly classified. Thus, the number of vertices that are reported as outliers is small.

Our local reconstruction algorithm for clusterable graphs is built upon our robust clustering oracle. That is, we first learn the cores of the input graph as before. Then (if the noisy part is not too large) we only "repair" all the vertices that are reported as outliers. Let v be any vertex that is reported as an outlier. We add all the neighbors of v in an explicit expander $G_{\exp}$ to "repair" the graph G, which is called a *hybridization* (between $G_{\exp}$ and G) and has been used to repair expander graphs in [KPS13]. Then the answers is guaranteed to be consistent with a graph G' such that its distance to the original graph G is at most d times the number of vertices that are reported as outliers, which has already been bounded to be small. In order to prove the claimed guarantee on cluster structure of G', we introduce a definition of *weak* vertices that intuitively correspond to the noisy part of the graph. Such a definition has also been used in [KPS13], though ours is more subtle, depending on the size of noise. We can show that one can improve

the cluster structure of the graph if we have repaired all the weak vertices in the above way. Furthermore, such weak vertices will always be reported as outliers, as guaranteed by our robust clustering oracle.

Organization. After giving the preliminaries in Section 2, we prove our two algorithmic results, i.e., Theorem 1.3 and Theorem 1.4, in Section 3 and 4, respectively, assuming that the local mixing property as stated in Theorem 1.5 holds. We give the proof of Theorem 1.5 in Section 5, and conclude in Section 6.

2 Preliminaries

Let $G = (V, E)$ denote an n-vertex undirected graph G with maximum degree bounded by some constant d, where $V = [n] := \{1, \cdots, n\}$. For each vertex v, we let d_v denote its degree. Throughout the paper, all the vectors will be row vectors unless otherwise specified or transposed to column vectors. For a vector $\mathbf{x}$, we let $\|\mathbf{x}\|_1 := \sum_i |\mathbf{x}(i)|$ and $\|\mathbf{x}\|_2 := \sqrt{\sum_i \mathbf{x}(i)^2}$ to denote its ℓ_1 norm and ℓ_2 norm, respectively. Let $\mathbf{1}_S$ denote the indicator vector of set S, that is $\mathbf{1}_S(u) = 1$ if $u \in S$ and 0 otherwise. Let $\mathbf{1}_v := \mathbf{1}_{\{v\}}$. Let $\mathcal{U}_S := \frac{\mathbf{1}_S}{|S|}$ denote the uniform distribution on set S. For any set X of vectors $\mathbf{x}_1, \cdots, \mathbf{x}_s$, we let $\text{span}(X) = \text{span}(\mathbf{x}_1, \cdots, \mathbf{x}_s)$ denote the linear span of X, that is $\text{span}(X) = \{\sum_{i=1}^s \mu_i \mathbf{x}_i | \mu_i \in \mathbb{R}\}$. For a vector $\mathbf{x}$ and a set S, we let $\mathbf{x}(S) := \sum_{v \in S} \mathbf{x}(S)$. For two distributions $\mathbf{p}_1$ and $\mathbf{p}_2$, we let $\|\mathbf{p}_1 - \mathbf{p}_2\|_{\text{TV}}$ denote the total variance distance between $\mathbf{p}_1, \mathbf{p}_2$. It is known that $\|\mathbf{p}_1 - \mathbf{p}_2\|_{\text{TV}} = \frac{1}{2}\|\mathbf{p}_1 - \mathbf{p}_2\|_1$.

Different types of random walks on G. We will consider the following random walks.

(1) *(Normal) random walk of t steps*. In a (normal) random walk, at each step, suppose we are at vertex v, then we jump to a random neighbor with probability $\frac{1}{2d}$ and stay at v with the remaining probability $1 - \frac{d_v}{2d}$. We stop the walk after t steps. We let $\mathbf{p}_v^t$ denote the probability vector for a t step random walk starting at v.

(2) *Uniform averaging walk of t steps*. In this walk, we choose a number $\ell \in \{0, 1, 2, \cdots, t-1\}$ uniformly at random, and stop the (normal) random walk after ℓ steps. We let $\mathbf{a}_v^t$ denote the probability vector for a uniform averaging walk of t steps starting at v.

(3) *Uniform averaging walk of t steps with two phases*. In this walk, we choose two integers $\ell_1, \ell_2 \in \{0, 1, 2, \cdots, t-1\}$ uniformly at random, and stop the walk after $\ell_1 + \ell_2$ steps. We let $\mathbf{b}_v^t$ denote the probability vector for a uniform averaging walk of t steps with two phases starting at v.

It is useful to note that for any two vertices u, v, $\mathbf{b}_u^t(v) = \sum_{w \in V} \mathbf{a}_u^t(w) \cdot \mathbf{a}_w^t(v)$.

Estimating reduced collision probabilities. Both our robust clustering oracle and local reconstruction needs to invoke a procedure to estimate the *reduced collision probability* of two random walks [KPS13]. For a vertex v, an integer t and a constant $\theta \in [0, 1]$, we let $S_v^\theta = \{u : \mathbf{a}_v^t(u) \leq \frac{1-\theta}{\sqrt{n}}\}$. For any two vertices u, v, the *θ-reduced collision probability* of u, v is defined as $\text{rcp}_\theta(u, v) = \sum_{w \in S_u^\theta \cap S_v^\theta} \mathbf{a}_u^t(w)\mathbf{a}_v^t(w)$.

Observe that by definition of $\mathbf{b}_v^t$-random walks, it holds that $\text{rcp}_0(u, v) \leq \sum_{w \in V} \mathbf{a}_v^t(w) \cdot \mathbf{a}_u^t(w) = \sum_{w \in V} \mathbf{a}_v^t(w) \cdot \mathbf{a}_w^t(u) = \mathbf{b}_v^t(u)$. The following lemma shows that under appropriate conditions, the reduced collision probability of two vertices can be well approximated in $\tilde{O}(\sqrt{n})$ time.

Lemma 2.1 ([KPS13]). *Let $\theta < \frac{1}{2}, \delta < 1$ be two constant. Let u, v be two vertices. There exists a procedure* ESTIMATERCP*$(G, u, v, \theta, \delta, t)$ that takes as input a d-bounded n-vertex graph G, vertices u, v, parameters θ, δ, and length parameter t, and satisfies the following properties:*

1. *It runs in time $O(\sqrt{n}t \log^2 n)$;*
2. *If $\boldsymbol{a}_u^t(S_u^\theta) \geq 1/2, \boldsymbol{a}_v^t(S_v^\theta) \geq 1/2$, then it aborts (without outputting an estimate) with probability at most $\exp(-\Theta(\sqrt{n}))$;*
3. *If it does not abort, then with probability at least $1 - \frac{1}{n^4}$, it outputs an estimate $\text{rcp}'(u, v)$ such that $\text{rcp}_\theta(u, v) - \delta \max\{\text{rcp}_\theta(u, v), \frac{1}{2n}\} \leq \text{rcp}'(u, v) \leq \text{rcp}_0(u, v) + \delta \max\{\text{rcp}_0(u, v), \frac{1}{2n}\}$.*

For the sake of completeness, we give the description of the algorithm ESTIMATERCP in Appendix B.2.

3 Robust Clustering Oracle

In this section, we present our algorithm for constructing the robust clustering oracle and answering the clustering queries. In the *preprocessing* (or *learning*) phase, the algorithm learns the cores (corresponding to clusters in the clusterable part) of the graph. In the *query* phase, the algorithm checks if the queried vertex v belongs to any of the learned cores or not to decide if it is an outlier or not. If not, the algorithm will find the index i corresponding to the cluster that v belongs to.

We will use the reduced collision probability of random walks of length $t = \frac{960 \log n}{\kappa \phi^2}$ for some sufficiently small constant $\kappa > 0$. Such probabilities can be efficiently estimated by invoking the ESTIMATERCP procedure (see Section 2). The intuition is that for most vertices v in a large cluster C, the uniform averaging walk of t steps from u will be close to the uniform distribution on C (by Theorem 1.5), which implies that for almost all of vertices $v \in C$, their reduced collision probability is at least $\frac{1-\kappa}{|C|}$.

The learning phase of the algorithm is as follows.

The preprocessing phase: LEARNCORE$(G, d, k, \phi, \varepsilon)$

1. Let θ_0 and δ_0 be two sufficiently small constant (say at most $\frac{1}{10^5}$). Let $\kappa > 0$ be a constant such that $\kappa = 100 \cdot \delta_0^2$. If $\varepsilon > \frac{\phi \kappa^2}{100}$, then abort and output **fail**.
2. Let $c > 0$ be a sufficiently large constant. Let $\tau_j = 3\sqrt{\frac{6\varepsilon}{\phi}}(1 + \frac{\kappa}{3})^j$ for $0 \leq j \leq J$, where $J := \arg\max_j \{\tau_j \leq 1\}$. (Note that $J = O(\frac{\log(\phi/\varepsilon)}{\kappa})$.) Let $t = \frac{960 \log n}{\kappa \phi^2}$.
3. Sample a set S of $\frac{c \cdot k^2 \ln k \cdot \log n}{\sqrt{\varepsilon/\phi}}$ vertices uniformly at random.
4. For any $u, v \in S$, run ESTIMATERCP$(G, u, v, \theta_0, \delta_0, t)$. If it does not abort then add an edge (u, v) with weight $\mathrm{rcp}'(u, v)$ in the *similarity graph* H on vertex set S.
5. Invoke FINDCORE(H, J) (to find *cores*).

The subroutine FINDCORE(H, J) is defined as follows.

FINDCORE(H, J)

1. Let $F = H$. Let $\mathcal{S} = \emptyset$.
2. For each $0 \leq j \leq J$, we iteratively do the following:
   (a) Let F_j denote the subgraph of F that consists of edges of weight at least $\frac{1-\kappa}{\tau_j}\frac{1}{n}$;
   (b) For each $v \in V(H)$: ▷ according to the lexicographical order of vertices
      i. Let $N(v)$ denote the neighborhood of v in F_j.
      ii. Find a maximal clique K from v by sequentially visiting all the edges incident to vertex v and all vertices $u \in N(v)$.
      iii. If a clique K with $|K| \geq (1 - \kappa)\tau_j |S|$ is found, then 1) add K to $\mathcal{S}$, and 2) remove all edges incident to K from F_j and F.
3. If $|\mathcal{S}| = 0$ or $|\mathcal{S}| > k$, then output **fail**; otherwise, output all the disjoint cliques (called *cores*), say $S_1, S_2, \cdots, S_h$, $h \leq k$, in $\mathcal{S}$.

Note that by the above definition of cores, it holds that for any core S_i, there exists $j_i \in \{0, 1, \cdots, J\}$ such that $\frac{|S_i|}{|S|} \geq (1 - \kappa)\tau_{j_i}$ and the edge weight in the clique $H[S_i]$ is at least $\frac{1-\kappa}{\tau_{j_i}}\frac{1}{n}$.

We need the following subroutine to answer clustering queries.

CHECKCORE(H, u)

1. Let $\kappa, \theta_0, \delta_0, t$ and τ_j be the same numbers as specified in the learning phase.
2. For any vertex $v \in S$, run ESTIMATERCP$(u, v, \theta_0, \delta_0, t)$.
   (a) If there exists a *unique* $i \leq h$ such that $\mathrm{rcp}'(u, v) \geq \frac{1-\kappa}{\tau_{j_i}} \frac{1}{n}$ for all $v \in S_i$, then return index i;
   (b) Otherwise, return **Outlier**.

Now we are ready to describe our algorithm for answering clustering queries.

The query phase:

ISOUTLIER(G, w):

1. If the learning phase outputs **fail**, then return **Yes**.
2. Otherwise, if CHECKCORE(H, w) returns **Outlier**, then return **Yes**.
3. Return **No**.

WHICHCLUSTER(G, w):

1. If ISOUTLIER(G, w) returns **Yes**, return **Outlier**.
2. Otherwise, return CHECKCORE(H, w).

SAMECLUSTER(G, x, y):

1. Run WHICHCLUSTER(G, x) and WHICHCLUSTER(G, y).
2. If none of the above two queries return **Outlier** and the returned two indices are identical, then output **Yes**.
3. Otherwise, return **No**.

3.1 The Analysis of Robust Clustering Oracle

In the following, we show the performance guarantee of the above algorithm. We will use the local mixing property on noisy clusterable graphs as guaranteed in Theorem 1.5, whose proof is deferred to Section 5. Recall from the description of our algorithm that $\kappa = 100 \cdot \delta_0^2$, which is a sufficiently small universal constant.

If $\varepsilon > \frac{\phi\kappa^2}{100}$ (i.e., the noise is too much), then by our algorithm, the learning phase will output **fail**. Any queried vertex will be reported as **Outlier**.

In the following, we assume that $\varepsilon \in [\Omega(\frac{\phi}{n}), \frac{\phi\kappa^2}{100}]$ and we prove the statement of Theorem 1.3. To do so, we first introduce the definition of strong vertices, which correspond to vertices in the clusterable part.

Definition and properties of strong vertices. Let $\phi \in (0, 1), \varepsilon \in [\Omega(\frac{\phi}{n}), \frac{\phi\kappa^2}{100}]$. Let G be an ε-perturbation of a $(k, \phi, \frac{a_{1.5}\varepsilon\kappa^4\phi}{3k^3 \log n})$-clusterable graph. Recall that $\mathbf{a}_v^t$ and $\mathbf{b}_v^t$ denote the distribution of the uniform average walk of length t and the uniform average walk of length t with two phases starting from v, respectively. In the algorithm, we invoke ESTIMATERCP with length parameter $t = \frac{960 \log n}{\kappa\phi^2}$.

We let $\varepsilon' := \frac{6\varepsilon}{\phi} < \frac{\kappa^2}{100}$. We introduce the following definition of strong vertex for the analysis, which was inspired by the corresponding definition for noisy expander graphs in [KPS13]. The main difference here is that we carefully take the size of clusters into consideration.

Definition 3.1. *We call a vertex v a* strong *vertex with respect to a subset C if $v \in C$, $|C| \geq 3\sqrt{\varepsilon'}n$ and $\|\boldsymbol{a}_v^t - \mathcal{U}_C\|_{TV} \leq \kappa$.*

Recall that θ_0 is small sufficiently small constant, $S_v^{\theta_0} = \{u : \mathbf{a}_v^t(u) \leq (1 - \theta_0)/\sqrt{n}\}$ and that $\mathrm{rcp}_{\theta_0}(u, v) = \sum_{w \in S_u^{\theta_0} \cap S_v^{\theta_0}} \mathbf{a}_u^t(w)\mathbf{a}_v^t(w)$ is the reduced collision probability of u, v (see Section 2). We have the following properties of strong/weak vertices, which easily follows from the proof of Lemma 2 in [KPS13]. We present the proof in Appendix C for the sake of completeness.

Lemma 3.2. *If a vertex u is strong with respect to a set C with $|C| \geq 3\sqrt{\varepsilon'}n$, then (1) there can be at most $\sqrt{\kappa}|C|$ vertices v in C with $\boldsymbol{a}_u^t(v) \leq (1-\sqrt{\kappa})/|C|$; (2) it holds that $\boldsymbol{a}_u^t(S_u^{\theta_0}) \geq 1/2$.*

Furthermore, if vertices u, v are both strong with respect to a set C with $|C| \geq 3\sqrt{\varepsilon'}n$, then we have that $\mathrm{rcp}_{\theta_0}(u,v) \geq (1-5\sqrt{\kappa})/|C|$.

The correctness of the robust clustering oracle. Now we show the correctness of the robust clustering oracle and bound the total number of vertices reported as outliers by the the algorithm. Recall that we let $P_i := \{u \in V : \textsc{WhichCluster}(u) = i\}$ with $1 \leq i \leq h$ for some integer h, and $B := \{u \in V : \textsc{IsOutlier}(u) = \textbf{Yes}\}$ denote the partition output by our algorithm.

Lemma 3.3. *Let G be an ε-perturbation of a $(k, \phi, \frac{a_{1.5}\varepsilon\kappa^4\phi}{3k^3\log n})$-clusterable graph. Then there exists a partition $D_1, \cdots, D_{h'}, B'$ for some $h' \leq k$ (that is independent of the order of queries), such that*

- *if $\varepsilon \in [\Omega(\frac{\phi}{n}), \frac{\phi}{60k^2}]$, then $h' \geq 1$ and each D_i is a $(\frac{\phi}{2}, \frac{a_{1.5}\sqrt{\varepsilon}\kappa^4\phi^{1.5}}{3k^3\log n})$-cluster, for any $1 \leq i \leq h'$; if $\varepsilon \in (\frac{\phi}{60k^2}, 1]$, then $h' = 0$; and*
- *with probability at least $1 - \frac{1}{n}$, the partition $P_1, \cdots, P_h, B$ output by the algorithm satisfies that $h' \leq h \leq k$ and $\sum_{i=1}^{h'} |P_i \triangle D_i| + \sum_{i=h'+1}^{h} |P_i| + |B| + |B'| \leq 40k\sqrt{\varepsilon/\phi}n$.*

In particular, the number of vertices reported as outliers is at most $40k\sqrt{\varepsilon/\phi}n$.

Proof. We first note that if $\varepsilon > \frac{\phi}{60k^2}$, then we can simply take $B' = V$ (and thus $h' = 0$) and then for any output partition of the algorithm, it holds that $\sum_{i=1}^{h'} |P_i \triangle D_i| + \sum_{i=h'+1}^{h} |P_i| + |B| + |B'| = 2n < 40k\sqrt{\varepsilon/\phi}n$.

Thus, in the following, we assume that $\varepsilon \leq \frac{\phi}{60k^2}$.

Let $\phi_{\text{out}} = \frac{a_{1.5}\varepsilon\kappa^4\phi}{3k^3\log n}$. Let $G^* = (V, E^*)$ be a $(k, \phi, \phi_{\text{out}})$-clusterable graph such that G is an ε-perturbation of G^*. Let $C_1, \cdots, C_{\overline{h}}$ be the corresponding $(\overline{h}, \phi, \phi_{\text{out}})$-clustering of G^* for some $\overline{h} \leq k$. That is, for each $i \leq \overline{h}$, $\phi_G(C_i) \leq \phi_{\text{out}}$, and one can insert/delete at most εdn edges inside subgraphs $G[C_i]$ to make all $G[C_i]$ become $(\phi, \phi_{\text{out}})$-clusters.

Now for each set C_i, we perform the following process on $G[C_i]$ recursively. We start with $B_i := \emptyset$ and $D_i := C_i$. If $|B_i| \leq \frac{|C_i|}{2}$, and there exists a subset $M_i \subseteq D_i$ with $|M_i| \leq |D_i|/2$ and $\phi_{G[C_i]}(M_i) \leq \phi/2$, then we update $B_i = B_i \cup M_i$, and $D_i = D_i \setminus M_i$. We recurse until no such set M_i can be found or $|B_i| > \frac{|C_i|}{2}$. Note that by our construction, the final set B_i satisfies that $\phi_{G[C_i]}(B_i) \leq \phi/2$ and that D_i has inner conductance at least $\phi/2$. Furthermore, it holds that $|B_i| \leq \frac{3}{4}|C_i|$, since right before the last update, we have that $|B_i'| \leq \frac{|C_i|}{2}$ and that the final cut M' satisfies that $|M_i'| \leq \frac{1}{2}(|C_i - B_i'|)$, which gives that $|B_i| \leq \frac{1}{2}(|C_i - B_i'|) + |B_i'| \leq \frac{3}{4}|C_i|$.

Now we claim that $|\cup_i B_i| \leq \frac{6\varepsilon}{\phi}n$. Assume on the contrary that $|\cup_i B_i| > \frac{6\varepsilon}{\phi}n$, i.e., $\sum_i |B_i| > \frac{6\varepsilon}{\phi}n$. First, we note that in order to make $\phi(G[C_i]) \geq \phi$, then we should add at least $\frac{\phi}{2}d\min\{|B_i|, |C_i - B_i|\} \geq \frac{\phi}{2}d \cdot \frac{1}{3}|B_i| = \frac{\phi}{6}d|B_i|$ edges, where the inequality follows from the fact that $|C_i - B_i| \geq \frac{1}{3}|B_i|$ which in turn is due to the fact that $|B_i| \leq \frac{3}{4}|C_i|$. Therefore, in order to make all C_i have inner conductance at least ϕ, we have to add at least $\sum_i \frac{\phi}{6}d|B_i| > \frac{\phi}{6}d \cdot \frac{6\varepsilon}{\phi}n = \varepsilon dn$ edges, which is a contradiction.

We note that since $\varepsilon \leq \frac{\phi}{60k^2}$, then it holds that at least one D_i has size at least $\frac{(1-(6\varepsilon/\phi))n}{k} \geq \frac{9n}{10k} \geq 3\sqrt{\frac{1}{10k^2}n} \geq 3\sqrt{\frac{6\varepsilon}{\phi}n} = 3\sqrt{\varepsilon'}n$. Now we apply Theorem 1.5 on G with error parameter $\varepsilon' = \frac{6\varepsilon}{\phi} < \frac{\kappa^2}{100}$, $\gamma = \frac{\kappa}{2}$, sets $C_i = D_i \cup B_i$, $1 \leq i \leq \overline{h}$ such that $\phi(G[D_i]) \geq \frac{\phi}{2}$, to obtain that for each D_i with $|D_i| \geq 3\sqrt{\varepsilon'}n$, there exists a subset $\widehat{D}_i \subseteq D_i$ such that $|\widehat{D}_i| \geq (1 - 4\sqrt{\varepsilon'})|D_i|$ and for any $v \in \widehat{D}_i$, and $t = \frac{960\log n}{\kappa\phi^2}$,

$$\|\mathbf{a}_v^t - \mathcal{U}_{C_i}\|_{\text{TV}} \leq \sqrt{\varepsilon'} + \frac{\kappa}{2} \leq \kappa.$$

This further implies that all vertices in $\widehat{D}_i$ are strong with respect to C_i, as $|C_i| \geq |D_i| \geq 3\sqrt{\varepsilon'}n$. We also note that for each D_i with $|D_i| \geq 3\sqrt{\varepsilon'}n$, it holds that $\phi_G(D_i) \leq \frac{\phi_{\text{out}} \cdot dn}{3\sqrt{\varepsilon'} \cdot dn} \leq \frac{a_{1.5}\sqrt{\varepsilon}\kappa^4\phi^{1.5}}{3k^3 \log n}$. Now we order D_i such that $|D_1| \geq \cdots \geq |D_{\overline{h}}|$ (breaking ties arbitrarily). Let h' be the largest index with $|D_{h'}| \geq 3\sqrt{\varepsilon'}n$. Note that $h' \geq 1$. We define the partition $D_1, \cdots, D_{h'}, B' := V \setminus (\cup_{i \leq h'} D_i)$. By definition, it holds that for each $1 \leq i \leq h'$, $|D_i| \geq 3\sqrt{\varepsilon'}n$ and $\phi(G[D_i]) \geq \frac{\phi}{2}, \phi_G(D_i) \leq \frac{\phi_{\text{out}} \cdot dn}{3\sqrt{\varepsilon'} \cdot dn} \leq \frac{a_{1.5}\sqrt{\varepsilon}\kappa^4\phi^{1.5}}{3k^3 \log n}$. Note that the partition $D_1, \cdots, D_{h'}, B'$ only depends on G. It holds that $|B'| = |\sum_i B_i| + |\cup_{i:|D_i|<3\sqrt{\varepsilon'}n} D_i| \leq (\varepsilon' + 3k\sqrt{\varepsilon'})n$.

We further define $D_g := \cup_{1 \leq i \leq h'} \widehat{D}_i$.

Now we show the following claim.

Claim 3.4. *With probability at least $1 - \frac{1}{n}$, for all vertices v in D_g,* WHICHCLUSTER*(v) will output a unique index $\sigma(i)$ if vertex $v \in \widehat{D}_i$ for some injection $\sigma : [h'] \to [k]$.*

Note that the statement of the lemma will then follow from the above claim: Let $h \leq k$ be the largest index output by the algorithm, and let $B, P_{\sigma(1)}, \cdots, P_{\sigma(h')}, P_j$, for $j \in [h] \setminus \{\sigma(1), \cdots, \sigma(h')\}$ be the partition output by the algorithm. Then by Claim 3.4, all vertices in D_g will be correctly partitioned and

$$\begin{aligned}
&\sum_{i=1}^{h'} |P_{\sigma(i)} \triangle D_i| + \sum_{j \in [h] \setminus \{\sigma(1), \cdots, \sigma(h')\}} |P_j| + |B| + |B'| \\
\leq\ & 2 \cdot (|\cup_{i:|D_i|<3\sqrt{\varepsilon'}n} D_i| + |\cup_{i:|D_i| \geq 3\sqrt{\varepsilon'}n} (D_i \setminus \widehat{D}_i)| + |\cup_i B_i|) \\
\leq\ & 2 \cdot (3k\sqrt{\varepsilon'} + 4\sqrt{\varepsilon'} + \varepsilon')n \\
\leq\ & 16k\sqrt{\varepsilon'}n \leq 40k\sqrt{\varepsilon/\phi}n.
\end{aligned}$$

Re-arranging the order of sets $D_1, \cdots, D_i$ will complete the proof of the Lemma. Now we prove the claim.

Proof of Claim 3.4. By the previous analysis, we have that for each i such that $|D_i| \geq 3\sqrt{\varepsilon'}n$, the number of vertices in C_i that are *not* strong (with respect to C_i) is at most

$$|D_i \setminus \widehat{D}_i| + |B_i| \leq 4\sqrt{\varepsilon'}|D_i| + \varepsilon' n \leq 5\sqrt{\varepsilon'}|D_i| \leq 5\sqrt{\varepsilon'}|C_i| \leq \frac{\kappa}{2}|C_i|.$$

That is, for each i such that $|C_i| \geq |D_i| \geq 3\sqrt{\varepsilon'}n$, at least $(1 - \frac{\kappa}{2})$ fraction of vertices in C_i are strong (with respect to C_i).

Now let us consider the sample set S. Recall that $|S| = \frac{c \cdot \log n}{\sqrt{\varepsilon/\phi}} = \Omega(\frac{\log n}{\sqrt{\varepsilon'}})$ for some large constant $c > 0$. Let $T_i = S \cap C_i$ and let $S'_i \subset T_i$ denote the set of vertices in T_i that are strong with respect to C_i. By Chernoff bound, we have that with probability at least $1 - 1/n^4$, for any i such that $|C_i| \geq |D_i| \geq 3\sqrt{\varepsilon'}n$,

$$\begin{aligned}
(1 - \frac{\kappa}{2})\frac{|C_i|}{n} \cdot |S| \ &\leq\ |T_i| \leq (1 + \frac{\kappa}{2})\frac{|C_i|}{n} \cdot |S|, \\
(1 - \kappa)\frac{|C_i|}{n} \cdot |S| < (1 - \frac{\kappa}{2})(1 - \frac{\kappa}{2})\frac{|C_i|}{n} \cdot |S| \ &\leq\ |S'_i| \leq (1 + \frac{\kappa}{2})\frac{|C_i|}{n} \cdot |S|.
\end{aligned}$$

In the following, we will condition on event that the above two inequalities hold.

Now recall that $\tau_j = 3\sqrt{\varepsilon'}(1 + \frac{\kappa}{2})^j$, for $0 \leq j \leq J$, where $J = O(\frac{\log(\phi/\varepsilon)}{\kappa})$ is the maximum integer j such that $\tau_j \leq 1$. Let j_i denote the index such that $|C_i| \in [\tau_{j_i} n, \tau_{j_i+1} n)$. Thus, $|S'_i| \geq (1 - \kappa)\tau_{j_i}|S|$.

Let v, u be two vertices in S'_i. By Lemma 3.2, we have that $\mathbf{a}_u^t(S_u^{\theta_0}) \geq 1/2, \mathbf{a}_v^t(S_v^{\theta_0}) \geq 1/2$, and $\text{rcp}_{\theta_0}(u, v) \geq (1 - 5\sqrt{\kappa})/|C_i|$. By the assumption that $\kappa = 100 \cdot \delta_0^2$ and Lemma 2.1, we obtain that with probability at least $1 - \frac{1}{n^4} - \exp(-\Theta(\sqrt{n}))$, ESTIMATERCP$(G, u, v, \theta_0, \delta_0, t)$ will output a value $\text{rcp}'(u, v)$

that is at least $(1-5\sqrt{\kappa})(1-\frac{\sqrt{\kappa}}{10})/|C_i| > \frac{1-\kappa/2}{|C_i|} > \frac{1-\kappa/2}{\tau_{j_i+1}n} \geq \frac{1-\kappa}{\tau_{j_i}n}$. That is, with probability at least $1-\frac{1}{n^3}$, in the similarity graph H, the induced subgraph $H[S_i']$ will form a complete graph with at least $(1-\kappa)\tau_{j_i}|S|$ vertices such that for each pair $u, v \in S_i'$, $\mathrm{rcp}'(u,v) \geq \frac{1-\kappa}{\tau_{j_i}n}$. Therefore, in our sample, the set S_i' will be recognized as a subgraph of a core (corresponding to C_i), which is a maximal clique with edge weight at least $\frac{1-\kappa}{\tau_{j_i}n}$.

Now once a vertex $v \in \widehat{D}_i$ is queried (for checking if it is outlier or not), then by using similar argument as above, we can guarantee that with probability at least $1-\frac{1}{n^3}$, for all $u \in S_i'$, the EstimateRCP will output $\mathrm{rcp}'(u,v)$ satisfying that $\mathrm{rcp}'(u,v) \geq \frac{1-\kappa}{\tau_{j_i}n}$. Thus, the algorithm will detect the core (corresponding to C_i) for v. Furthermore, for any vertex v that is strong with respect to C_i, it holds that for any C_j with $j \neq i$, there can be at most $\kappa|C_j|$ vertices $u \in C_j$ with $\mathrm{rcp}_{\theta_0}(u,v) > \frac{1-5\sqrt{\kappa}}{|C_j|}$, this is true since the total probability mass on C_j of the random walk distribution from v is at most κ. This ensures that there will be a unique core corresponding to v. Let $\sigma : [h'] \to [h']$ denote the corresponding bijection between $\{C_1, \cdots, C_{h'}\}$ and the cores $\{S_1, \cdots, S_{h'}\}$ found by the algorithm. By union bound, we have that with probability at least $1-\frac{1}{n}$, for each strong vertex $v \in S_i'$, the algorithm will answer the corresponding index $\sigma(i)$ to the query WhichCluster(v). ∎

Running time and query complexity. Note that in the learning phase, we need to invoke the procedure EstimateRCP for $|S| \times |S| = O(\frac{k^4 \ln^2 k \cdot \phi \log^2 n}{\varepsilon})$ times, and each invocation takes time $O(\sqrt{n}t \log^2 n)$, which in total takes time $O(\sqrt{n}\frac{\log n}{\phi^2} \cdot \frac{k^4 \ln^2 k \cdot \phi \log^2 n}{\varepsilon}) = O(\sqrt{n}\frac{k^4 \ln^2 k \cdot \log^3 n}{\phi\varepsilon})$. Finding the cores in the similarly graph can be implemented by a simple greedy algorithm, which can be implemented in $O(\mathrm{poly}(|S|)) = O(\mathrm{poly}(\frac{k \cdot \phi \log n}{\varepsilon}))$ time. Thus, the query complexity and running time in the learning phase is dominated by $O(\sqrt{n} \cdot \mathrm{poly}(\frac{k \cdot \log n}{\varepsilon\phi}))$, which, by similar arguments, also upper bounds the query complexity and running time on each query vertex w in the query phase.

Remark. From Lemma 3.3 and its proof, we note that in order to guarantee that $h' \geq 1$, i.e., there exists at least one good cluster D_i, we need to set $\varepsilon = O(\frac{\phi}{k^2})$ (so that there exists at least one set with size at least $(1-\varepsilon')n/k \geq 3\sqrt{\varepsilon'}n$). Thus our algorithm has non-trivial guarantee only if the adversary does not perturb the graph too much. Suppose that there are $h \leq k$ ground-truth clusters $C_1, \cdots, C_h$ and the adversary perturbs an ε-fraction on intra-cluster edges. In order to recover for *each* C_i, a subset P_i that is close to $D_i \subseteq C_i$, then we need to require that $\min_{i \in h} |D_i| \geq 3\sqrt{\varepsilon'}n$, which can be satisfied if $\varepsilon = O(\phi \cdot (\frac{\min_{i \in h}|D_i|}{n})^2)$.

We further remark that in our setting, any algorithm can only be able to (partially) recover the large clusters, say of size at least $\Omega(\varepsilon n)$. This is the case as for any small cluster (of size $o(\varepsilon n)$), it can be completely hidden or destroyed by the adversary. Currently, our analysis shows that our algorithm can recover the cluster of size $\Omega(\sqrt{\frac{\varepsilon}{\phi}}n)$. It will be an interesting question to design a robust clustering oracle that can recover smaller clusters (i.e., of size in the range $[\Omega(\varepsilon n), o(\sqrt{\frac{\varepsilon}{\phi}}n)]$).

4 The Local Reconstruction Algorithm

In this section, we present our reconstruction algorithm, which will be built upon our robust clustering oracle algorithm in Section 3 and consists of two phases: the *learning* phase, that learns the cores (corresponding to clusters in the clusterable part) of the graph, and the *query* phase, which first checks if the queried vertex belongs to any of the learned cores or not, and then output its neighbors in the amended clusterable graph accordingly. We need the following tool of explicit construction of expanders.

Explicit expanders. For any vertex set $V = [n]$, we let $G_{\exp} = (V, E_{\exp})$ denote a graph on V with maximum degree at most 16 such that for any set S in $G_{\exp}$ with $|S| \leq n/2$, it holds that $|E_{\exp}(S, V \setminus S)| \geq \eta|S|$, for some constant $\eta > 0$. It is known (see e.g., Lemma 6 in [KPS13] which builds upon [GG81]) that such an expander $G_{\exp}$ also exists and can be *explicitly* constructed in the sense that for any specified vertex v, one can find all neighbors of v in $G_{\exp}$ in $\text{poly}(\log n)$ time.

In the following, given a graph G, we let $G_{\exp}$ denote an explicit expander graphs on the same vertex set as G. We call vertices G or $G_{\exp}$-neighbors of a vertex v, depending on the graph under consideration.

Local reconstruction:

1. Run LEARNCORE$(G, d, k, \phi, \varepsilon)$.
2. For each query NEWNEIGHBORS(G, w):
   (a) If the learning phase outputs **fail**, then output all $G_{\exp}$-neighbors and G-neighbors of w.
   (b) Otherwise, run CHECKCORE(H, w).
      i. If w is reported as **Outlier**, then add all the $G_{\exp}$-edges (w, u) incident to w;
      ii. Otherwise, for each vertex u that is a $G_{\exp}$-neighbor of w, run CHECKCORE(H, u). If u is reported as **Outlier**, then add edge (w, x).

   Output all neighbors added to w and all G-neighbors of w.

Note that the algorithm should be implemented by first taking as input a random seed s, which is fixed once for all (and used for sampling vertices in the learning phase and performing random walks), and then on any query vertex v, *deterministically* outputting the neighborhood of v in the graph G'. By construction, if an edge (u, v) is added, then on query vertex u, v will be output as a neighbor of u and vice versa. Therefore, the algorithm is independent of the order of queries and the answer will be globally consistent.

4.1 Analysis of the Local Reconstruction Algorithm

In the following, we show the performance guarantee of the above algorithm and prove Theorem 1.4. We first note that the running time and query complexity can be analyzed in the same way as in the proof Theorem 1.3.

It follows from the definition of $G_{\exp}$ that the maximum degree of G' is bounded by $d + 16$, as $G_{\exp}$ has maximum degree at most 16 and for each vertex u that is found to be an outlier, we will add all of its $G_{\exp}$-neighbors to u.

Recall from the description of our algorithm that $\kappa > 0$ is a sufficiently small universal constant. If $\varepsilon > \frac{\phi\kappa^2}{100}$ (i.e., the noise is too much), then by our algorithm, the learning phase will output **fail**. Furthermore, on query any vertex u, the query phase will output all of its G and $G_{\exp}$ neighbors of u. Thus, G' is a complete hybridization of G and $G_{\exp}$. Note that for any set $S \subset V$, $|E'(S, \bar{S})| \geq |E_{\exp}(S, \bar{S})|$, where E' and $E_{\exp}$ denote the set of edges in G' and $G_{\exp}$ respectively. Thus, it holds that if $|S| \leq \frac{n}{2}$, $\phi_{G'}(S) = \frac{|E_{\exp}(S,\bar{S})|}{d|S|} \geq \frac{\eta}{d}$, where we used the fact that for any set S with $|S| \leq \frac{n}{2}$ in $G_{\exp}$, $|E_{\exp}(S, \bar{S})| \geq \eta|S|$. Therefore, the resulting graph G' is $(1, \frac{\eta}{d}, 0)$-clusterable. Furthermore, the number of edges added to G is at most $16n/2 = 8n = O(\min\{1, k\sqrt{\varepsilon/\phi}\} \cdot n)$ as $\varepsilon > \frac{\phi\kappa^2}{100}$. Thus, in this case, the statement of our theorem holds.

In the following, we prove the rest properties as listed in Theorem 1.4 for the more interesting case that $\varepsilon \in [\Omega(\frac{\phi}{n}), \frac{\phi\kappa^2}{100}]$.

In this case, the description of the local reconstruction algorithm, the number of added edges is 16 times the number of vertices that are reported as outliers, and thus by Lemma 3.3, is at most $16 \times 40k\sqrt{\frac{\varepsilon}{\phi}}n = 640k\sqrt{\frac{\varepsilon}{\phi}}n$. Now we analyze the cluster structure of the resulting graph.

Definition and property of weak vertices. Let $\varepsilon' := \frac{6\varepsilon}{\phi} < \frac{\kappa^2}{100}$. We introduce the following definitions of weak vertex for the analysis, which was inspired by the corresponding definitions for noisy expander graphs in [KPS13]. The main difference here is that we carefully take the size of clusters into consideration.

Definition 4.1. *We call a vertex v* weak *vertex, if for any subset A with $|A| \geq \frac{2\varepsilon'}{3}n$, it holds that $\|\boldsymbol{b}_v^t - \mathcal{U}_A\|_{TV} \geq 1/4$.*

In order to analyze the cluster structure of the resulting graph G', we need the following property of weak vertices.

Lemma 4.2. *With probability at least $1 - n^{-3}$, it holds that for any weak vertex u, the algorithm will report u as an outlier.*

Proof. We first show that if u is weak, then for any subset A with $|A| \geq \frac{2\varepsilon'}{3}n$ vertices, at most $7/8|A|$ vertices v in A satisfy $\mathbf{b}_u^t(v) \geq \frac{7/8}{|A|}$. This is true since otherwise, there will be more than $7/8|A|$ vertices v satisfy $\mathbf{b}_u(v) \geq \frac{7/8}{|A|}$. If we let $A_1 \subseteq A$ (resp. $A_2 \subseteq A$) denote the set of vertices v in A such that $\mathbf{b}_u^t(v) \leq \frac{1}{|A|}$ (resp. $\mathbf{b}_u^t(v) > \frac{1}{|A|}$), then

$$\begin{aligned}
\|\mathbf{b}_u^t - \mathcal{U}_A\|_{TV} &= \frac{1}{2}\left(\sum_{v\in A_1}(\frac{1}{|A|} - \mathbf{b}_u^t(v)) + \sum_{v\in A_2}(\mathbf{b}_u^t(v) - \frac{1}{|A|}) + \sum_{v\in V\setminus A}\mathbf{b}_u^t(v)\right) \\
&= \sum_{v\in A_1}(\frac{1}{|A|} - \mathbf{b}_u^t(v)) \\
&< (1-7/8)|A|\cdot\frac{1}{|A|} + |A|\cdot\frac{1-7/8}{|A|} = 2(1-7/8) < \frac{1}{4},
\end{aligned}$$

which is a contradiction. By the definitions of reduced collision probability $\mathrm{rcp}_{\theta_0}(u,v)$ and relations of $\mathbf{a}_u^t$ and $\mathbf{b}_u^t$, we have that $\mathrm{rcp}_0(u,v) \leq \mathbf{b}_u^t(v)$, and thus there can be at most $\frac{7}{8}|A|$ vertices v in A with $\mathrm{rcp}_0(u,v) \geq \frac{7}{8|A|}$. Note that this property holds for all sets A with $|A| \geq \frac{2\varepsilon'}{3}n$.

For each $0 \leq j \leq J$, we let T_j denote the set of vertices v such that $\mathrm{rcp}_0(u,v) \geq \frac{7}{8\tau_j n}$. Recall that $\tau_j = 3\sqrt{\varepsilon'}(1+\frac{\kappa}{2})^j$, for $0 \leq j \leq J$.

If $|T_j| \geq \tau_j n > \frac{2\varepsilon'}{3}n$, then for all vertices $v \in T_j$, it holds that $\mathrm{rcp}_0(u,v) \geq \frac{7}{8\tau_j n} \geq \frac{7}{8|T_j|}$, which is a contradiction. If $\frac{7\tau_j}{8}n < |T_j| < \tau_j n$, then we can add arbitrarily at most $\frac{1}{8}\tau_j n$ vertices to T_j to obtain a set A such that $|A| = \tau_j n > \frac{2\varepsilon'}{3}n$, and for at least $\frac{|T_i|}{|A|} > \frac{7}{8}$ fraction of vertices v in A, it holds that $\mathrm{rcp}_0(u,v) \geq \frac{7}{8\tau_j n} = \frac{7}{8|A|}$, which is a contradiction. Therefore, it must hold that $|T_j| \leq \frac{7\tau_j n}{8}$.

That is, for the weak vertex u, it holds that for each $0 \leq j \leq J$, there will be at most $\frac{7}{8}\tau_j n$ vertices v with $\mathrm{rcp}_0(u,v) \geq \frac{7}{8\tau_j n}$. Thus, there will be at least $(1-\frac{7}{8}\tau_j)n$ vertices v with $\mathrm{rcp}_0(u,v) \leq \frac{7}{8\tau_j n}$. We can further guarantee that with probability at least $1-1/n^2$, for any such pair u, v, the procedure ESTIMATERCP (with parameter $\delta \leq \frac{\sqrt{\kappa}}{10}$) either aborts or outputs an estimate $\mathrm{rcp}'(u,v) \leq (1+\frac{\sqrt{\kappa}}{10})\frac{7}{8\tau_j n} \leq \frac{8}{9\tau_j n}$, for any $0 \leq j \leq J$. Finally, with probability at least $1 - \frac{2}{n^2}$, in our sample set S, at least $(1-\frac{7}{8}\tau_j)$ fraction of vertices v satisfy that $\mathrm{rcp}'(u,v) \leq \frac{8}{9\tau_j n}$, or equivalently, less than $\frac{7}{8}\tau_j$ fraction of vertices v satisfy that $\mathrm{rcp}'(u,v) \geq \frac{8}{9\tau_j n}$. This implies that our algorithm will report u as an outlier. ∎

Cluster structure of G'. Now we are ready to show that the resulting graph G' from our local reconstruction algorithm can be partitioned into at most k parts, each of which has relatively large inner conductance.

Lemma 4.3. *Let $\phi^* = \frac{a_{4.3}\varepsilon\phi}{k^4 \log n}$ for some sufficiently small constant $a_{4.3}$. If G is an ε-perturbation of a $(k, \phi, \frac{a_{1.5}\varepsilon\kappa^4\phi}{3k^3 \log n})$-clusterable graph, then the resulting graph G' from the local reconstruction algorithm is $(k, \phi^*, 1)$-clusterable.*

Proof. For analysis, we perform the following procedure on the input graph G. Let $\gamma = \frac{\varepsilon'}{3} = \frac{2\varepsilon}{\phi}$. We start with the set $U := V$ and a partitioning $\mathcal{P} := \{V\}$ of G. Then if there exists a set $U \in \mathcal{P}$ and $S \subseteq U$ such that $\gamma n \leq |S| \leq \frac{|U|}{2}$ and $\phi_G(S) \leq \phi_{\text{out}} := \frac{a_{1.5}\varepsilon\kappa^4\phi}{3k^3 \log n}$, then we set $\mathcal{P} = (\mathcal{P} \setminus \{U\}) \cup \{S, U \setminus S\}$. We repeat until no such S can be found. Let $\mathcal{P} = \{C_1, \cdots, C_h\}$ denote the final partitioning of V.

Note that for any C_i, if U is the subset that contains C_i and is then split into C_i and $U \setminus C_i$, then $|U| \geq 2\gamma n$ and thus $|C_i| \geq \gamma n$ and $|U \setminus C_i| \geq \frac{|U|}{2} \geq \gamma n$ by the construction. This implies that at the end of the above procedure, it holds that $\min_i |C_i| \geq \gamma n$.

We further note that $|\mathcal{P}| = h \leq k$. This is true since otherwise, in order to make G become a $(k, \phi, \phi_{\text{out}})$-clusterable graph, one has to patch up at least one set C_i to other parts, that is, we need to add at least $\frac{3\phi}{4} \cdot d \min_i\{|C_i|\} \geq \frac{3\phi}{4} \cdot d \cdot \frac{2\varepsilon}{\phi} n > \varepsilon dn$ edges, which is a contradiction to the assumption that G is an ε-perturbation of a $(k, \phi, \phi_{\text{out}})$-clusterable graph.

Now let us consider the partition $\mathcal{P}$ in the constructed graph G'. Observe that by the description of our algorithm, for any set S of vertices $|E'(S, \bar{S})| \geq |E(S, \bar{S})|$, where E' and E denote the set of edges in G' and G respectively. In particular, Lemma 4.2 implies that the set of G'-neighbors of any weak vertex u is a superset of the set of G-neighbors of u, as u will be reported as an outlier by the algorithm and the G_{exp}-neighbors of u will be added to G'.

We have the following claim.

Claim 4.4. *In the graph G', for each C_i, and any subset $S \subset C_i$ with $|S| \leq \frac{|C_i|}{2}$, it holds that $\phi_{G'}(S) \geq k\phi^*$.*

Proof. If $\gamma n \leq |S| \leq \frac{|C_i|}{2}$, then by our construction of C_i, we have that $\phi_G(S) \geq \phi_{\text{out}}$. Thus, $\phi_{G'}(S) = \frac{|E'(S, V\setminus S)|}{d|S|} \geq \frac{|E(S, V\setminus S)|}{d|S|} \geq \phi_{\text{out}} \geq k\phi^*$. Now let us consider the case that $|S| \leq \gamma n$.

If there are less than $(1 - \frac{\eta}{2})$ fraction of vertices in S are weak, then we show that $\phi_G(S) \geq k\phi^*$. Suppose this is not the case, that is, $\phi_G(S) < \frac{a_{4.3}\varepsilon\phi}{k^3 \log n} \leq \frac{\eta}{16t}$, if we set $a_{4.3}$ to be a sufficiently small constant. By the proof of Theorem 4 in [KPS13] (which in turn is based on the proof of Lemma 4.7 in [CS10]), we know that for at least $(1 - \eta/2)$ fraction of vertices u in S, the probability that a $\mathbf{b}_u^t$-random walk that starts at u will end up in $\bar{S}$ is at most $1/4$. Now let A be any set with $|A| \geq \frac{2\varepsilon'}{3} n$. Since $|S| \leq \frac{\varepsilon'}{3} n$, it holds that $|A \setminus S| \geq \frac{1}{2}|A|$. Thus, we have that $\mathcal{U}_A(A \setminus S) \geq \frac{1}{2}$. This gives that $\|\mathbf{b}_u^t - \mathcal{U}_A\|_{\text{TV}} \geq \frac{1}{4}$, which implies that such a vertex u is weak. Thus, S contains at least $(1 - \eta/2)$ fraction of weak vertices, which is a contradiction. This implies that $\phi_{G'}(S) \geq \phi_G(S) \geq k\phi^*$.

If there are more than $(1 - \eta/2)$ fraction of weak vertices, denoted by W, in S, then the number of G_{exp}-neighbors of W in G_{exp} is at least $\eta|W|$. Since all these G_{exp}-neighbors are also in G', we have that the number of vertices outside of S is at least $\eta|W| - |S \setminus W| \geq \eta(1 - \eta/2)|S| - \eta/2|S| \geq \frac{\eta}{6}|S|$. Since we add all the edges in G_{exp} that are incident to W to G', we have that the number of edges crossing S in G' is at least $\frac{\eta}{6}|S|$, and thus $\phi_{G'}(S) \geq \frac{\eta}{6d} \geq k\phi^*$. ∎

Now based on the partition $\mathcal{P} = \{C_1, \cdots, C_h\}$ as constructed above, we find a new partition of G' such that each part has large inner conductance. We start with the partition $\mathcal{P} = \{C_1, \cdots, C_h\}$ as constructed above and perform the following operations. If there exist $i, j \leq h$, $S \subseteq C_i$ satisfies that $i \neq j$, $|S| \leq \frac{|C_i|}{2}$ and that $|E'(S, C_i \setminus S)| < |E'(S, C_j)|$, then we set $C_i := C_i \setminus S$ and $C_j := T_j \cup S$. We repeat until the condition is violated.

Note that the above process always terminates in a finite number of steps since the number of crossing edges, i.e., $\sum_{i \neq j} |E'(C_i, C_j)|$, always decreases in each iteration. Furthermore, we observe that at the end

of the process, for any $1 \leq i \leq h$, and any set $S \subseteq C_i$ with $|S| \leq \frac{|C_i|}{2}$, $|E'(S, C_i \setminus S)| \geq \frac{|E'(S,V\setminus S)|}{k}$. Therefore, $\phi_{G'[C_i]}(S) \geq \frac{1}{k}\phi_{G'}(S) \geq \phi^*$. This implies that for each i, $\phi(G'[C_i]) \geq \phi^*$. ∎

5 Local Mixing Property of Random Walks on Noisy Clusterable Graphs: Proof of Theorem 1.5

In this section, we give the proof of Theorem 1.5. To do so, we first give a property of random walks on clusterable graphs (without noise).

5.1 Local Mixing Property of Random Walks on Clusterable Graphs

We will first prove a mixing property of random walks on a clusterable graph, which says that in a clusterable graph, for many vertices v in a large cluster, a random walk of appropriate length starting from v will mix well inside the corresponding cluster. By a simple reduction (see Appendix B), it suffices to consider a corresponding weighted d-regular graph for any d-bounded graph.

Theorem 5.1. *Let $0 < \alpha, \beta, \xi \leq 1$. Let $\phi_{out} \leq a_{5.1}\frac{\xi\alpha\beta\phi_{in}^2}{k^3 \log n}$ for some sufficiently small constant $a_{5.1} > 0$. Let G be a weighted d-regular and $(k, \phi_{in}, \phi_{out})$-clusterable graph with underlying clusters $C_1, \cdots, C_h$ for some $h \leq k$. Then for each C_i with $|C_i| \geq \alpha n$, there exists a subset $C_i' \subseteq C_i$ such that $|C_i'| \geq (1-\beta)|C_i|$, and for any $v \in C_i'$, and $t = \frac{20 \log n}{\phi_{in}^2}$, it holds that*

$$\|\boldsymbol{p}_v^t - \mathcal{U}_{C_i}\|_{TV} \leq \xi.$$

We remark that [ST13] and [AOPT16] gave analysis for upper bounding the probability that a random walk of length t from a typical vertex v in a set S with small conductance will escape the set S, and lower bounding the probability that the walk from v of length t stays inside S, respectively. It is unclear if one can use their analysis to prove the above theorem. In the following, we prove Theorem 5.1 by using some strong spectral property of clusterable graphs, i.e., the spectral gap between λ_{h+1} and λ_h for some $h \leq k$, and the closeness of the space spanned by the first h eigenvectors and the space spanned by the indicator vectors of clusters. More precisely, we need the following tools.

Lemma 5.2 (Lemma 5.2 in [CPS15] and Lemma 10 in [CKK+18])**.** *Let G be a weighted d-regular and $(k, \phi_{in}, \phi_{out})$-clusterable graph with underlying clusters $C_1, \cdots, C_h$ for some $h \leq k$. Then $\lambda_h \leq 2\phi_{out}$ and $\lambda_{h+1} \geq \frac{\phi_{in}^2}{2}$.*

Fact 5.3. *It holds that $\|\boldsymbol{1}_v\|_2^2 = \sum_{j=1}^n \mathbf{v}_j(v)^2 = 1$, for any $v \in V$.*

The following is a direct corollary of a structural result due to [PSZ17] that relates the first k eigenvectors of the Laplacian to the normalized indicator vectors of some k-partition of the graph. Recall that $\mathbf{v}_i$ is the eigenvector corresponding to the i-th smallest eigenvalue of the Laplacian of G.

Theorem 5.4. *Let $\phi_{out} \leq a_{5.4}\phi_{in}^2/k^2$ for sufficiently small constant $a_{5.4} > 0$. Let G be a weighted d-regular and $(k, \phi_{in}, \phi_{out})$-clusterable graph with underlying (ϕ_{in}, ϕ_{out})-clusters $C_1, \cdots, C_h$ for some $h \leq k$. Let $\mathbf{r}_i := \frac{1}{\sqrt{|C_i|}} \cdot \boldsymbol{1}_{C_i}$. Then there exist h orthonormal vectors $\tilde{\mathbf{r}}_1, \cdots, \tilde{\mathbf{r}}_h \in$ span$(\mathbf{r}_1, \cdots, \mathbf{r}_h)$ and a constant $c_{5.4} > 0$, such that*

$$\|\mathbf{v}_i - \tilde{\mathbf{r}}_i\|_2^2 \leq c_{5.4} \cdot \frac{h\phi_{out}}{\phi_{in}^2}.$$

Proof. Let $\rho(h) := \min_{A_1,\cdots,A_h} \max\{\phi_G(A_i) : i = 1, \cdots, h\}$, where the minimum is taken over all h-partitions $A_1, \cdots, A_h$. It is proven in Theorem 1.1 of [PSZ17] that if $\lambda_{h+1}/\rho(h) \geq ch^2$ for some constant $c > 0$, then there exist orthonormal vectors $\tilde{\mathbf{r}}_1, \cdots, \tilde{\mathbf{r}}_h \in$ span$(\mathbf{r}_1, \cdots, \mathbf{r}_h)$ such that $\|\mathbf{v}_i - \tilde{\mathbf{r}}_i\|_2^2 \leq 1.1h \cdot \frac{\rho(h)}{\lambda_{h+1}}$.

Note that by definition, $\rho(h) \leq \phi_{\text{out}}$. In addition, by Lemma 5.2, it holds that $\lambda_{h+1} \geq \frac{\phi_{\text{in}}^2}{2}$. Furthermore, since $\phi_{\text{out}} \leq a_{5.4}\phi_{\text{in}}^2/k^2 \leq a_{5.4}\phi_{\text{in}}^2/h^2$, it holds that $\lambda_{h+1}/\rho(h) \geq \frac{\phi_{\text{in}}^2}{2\phi_{\text{out}}} = ch^2$ as $a_{5.4}$ is sufficiently small constant. This then implies that $\|\mathbf{v}_i - \tilde{\mathbf{r}}_i\|_2^2 \leq 1.1h \cdot \frac{\rho(h)}{\lambda_{h+1}} \leq c_{5.4} \cdot \frac{h\phi_{\text{out}}}{\phi_{\text{in}}^2}$ for some constant $c_{5.4}$. ∎

Now we are ready to prove Theorem 5.1. We first provide a high level idea. We will bound the ℓ_2-norm distance of the random walk distribution $\mathbf{p}_v^t$ and the uniform distribution $\mathcal{U}_C$ over the cluster C that contains v, i.e., $\|\mathbf{p}_v^t - \mathcal{U}_C\|_2$. In order to do so, we note that by Theorem 5.4, the vector $\mathcal{U}_C$, which is a scale of the indicator vector of C, lies in a space that can be well approximated by the space of the first h (where $h \leq k$ is the number of clusters) eigenvectors of matrix $\mathbf{P}$. Using this, we show that the projection of $\mathbf{p}_v^t - \mathcal{U}_C$ on the space spanned by the first h eigenvectors is small. Furthermore, by Lemma 5.2, λ_{h+1} is large, and thus the length of the projection of $\mathbf{p}_v^t - \mathcal{U}_C$ on the space spanned by the remaining $n-h$ eigenvectors is dominated by $(1 - \frac{\lambda_{h+1}}{2})^{O(t)}$, which is also small for appropriately chosen t. Now we give the details.

Proof of Theorem 5.1. For any vertex v, we let $X_v := \sum_{j=1}^h \mathbf{v}_j(v)^2$. We first note that $\sum_{v\in V} X_v = \sum_{v\in V}\sum_{j=1}^h \mathbf{v}_j(v)^2 = \sum_{j=1}^h \|\mathbf{v}_j\|_2^2 = h$. Therefore, by the averaging argument, there can be at most $\frac{\beta\alpha}{2}n$ vertices v with $X_v \geq \frac{2h}{\beta\alpha n}$.

Note that by the precondition of the Theorem, it holds that $\phi_{\text{out}} \leq a_{5.4}\phi_{\text{in}}^2/k^2$. Let $\mathbf{r}_i$ and $\tilde{\mathbf{r}}_i$ be the vectors as defined in Theorem 5.4. Let $Y_v := \sum_{j=1}^h (\mathbf{v}_j(v) - \tilde{\mathbf{r}}_j(v))^2$. Then by applying Theorem 5.4 with graph G, we have that

$$\sum_v Y_v = \sum_v \sum_{j=1}^h (\mathbf{v}_j(v) - \tilde{\mathbf{r}}_j(v))^2 = \sum_{j=1}^h \|\mathbf{v}_j - \tilde{\mathbf{r}}_j\|_2^2 \leq c_{5.4} \cdot \frac{h^2\phi_{\text{out}}}{\phi_{\text{in}}^2}$$

Again, by the averaging argument, there can be at most $\frac{\beta\alpha}{2}n$ vertices v with $Y_v \geq c_{5.4} \cdot \frac{h^2\phi_{\text{out}}}{\phi_{\text{in}}^2}\frac{2}{\beta\alpha n}$.

Now let us define $C_i' := \{v : v \in C_i, X_v \leq \frac{2h}{\beta\alpha n}, Y_v \leq c_{5.4} \cdot \frac{h^2\phi_{\text{out}}}{\phi_{\text{in}}^2}\frac{2}{\beta\alpha n}\}$. Note that for any C_i with $|C_i| \geq \alpha n$, it holds that $|C_i'| \geq |C_i| - (\frac{\beta\alpha}{2} + \frac{\beta\alpha}{2})n \geq |C_i| - \beta|C_i| \geq (1-\beta)|C_i|$.

Let us consider any vertex $v \in C_i'$. Since $\mathbf{r}_i = \frac{\mathbf{1}_{C_i}}{\sqrt{|C_i|}}$, it holds that

$$\mathcal{U}_{C_i} = \frac{\mathbf{1}_{C_i}}{|C_i|} = \langle \mathbf{1}_v, \frac{\mathbf{1}_{C_i}}{\sqrt{|C_i|}}\rangle \cdot \frac{\mathbf{1}_{C_i}}{\sqrt{|C_i|}} = \langle \mathbf{1}_v, \mathbf{r}_i\rangle \cdot \mathbf{r}_i = \sum_{j=1}^h \mathbf{r}_j(v) \cdot \mathbf{r}_j \quad = \quad \sum_{j=1}^h \tilde{\mathbf{r}}_j(v) \cdot \tilde{\mathbf{r}}_j$$

where the last equation follows from the fact that $\tilde{\mathbf{r}}_1, \cdots, \tilde{\mathbf{r}}_h$ have the same linear span as vectors $\mathbf{r}_1, \cdots, \mathbf{r}_h$, which in turn follows from the properties of $\{\tilde{\mathbf{r}}_i\}$ as guaranteed by Theorem 5.4.

Recall that $\mathbf{p}_v^t = \sum_{j=1}^n (1 - \frac{\lambda_j}{2})^t \mathbf{v}_j(v) \cdot \mathbf{v}_j$. We let $t = \frac{20\log n}{\phi_{\text{in}}^2}$. Thus, we have that

$$\begin{aligned}
&\|\mathbf{p}_v^t - \mathcal{U}_{C_i}\|_2 = \|\sum_{j=1}^n (1 - \frac{\lambda_j}{2})^t \mathbf{v}_j(v) \cdot \mathbf{v}_j - \sum_{j=1}^h \tilde{\mathbf{r}}_j(v) \cdot \tilde{\mathbf{r}}_j\|_2 \\
= \;& \|\sum_{j=1}^n (1 - \frac{\lambda_j}{2})^t \mathbf{v}_j(v) \cdot \mathbf{v}_j - \sum_{j=1}^h \mathbf{v}_j(v) \cdot \mathbf{v}_j + \sum_{j=1}^h \mathbf{v}_j(v) \cdot \mathbf{v}_j - \sum_{j=1}^h \tilde{\mathbf{r}}_j(v) \cdot \tilde{\mathbf{r}}_j\|_2 \\
\leq \;& \|\sum_{j=1}^h ((1 - \frac{\lambda_j}{2})^t - 1)\mathbf{v}_j(v) \cdot \mathbf{v}_j\|_2 + \|\sum_{j=h+1}^n (1 - \frac{\lambda_j}{2})^t \mathbf{v}_j(v) \cdot \mathbf{v}_j\|_2 + \|\sum_{j=1}^h \mathbf{v}_j(v) \cdot \mathbf{v}_j - \sum_{j=1}^h \tilde{\mathbf{r}}_j(v) \cdot \tilde{\mathbf{r}}_j\|_2
\end{aligned}$$

$$
\begin{aligned}
&\leq \sqrt{\sum_{j=1}^{h}((1-\frac{\lambda_j}{2})^t-1)^2\mathbf{v}_j(v)^2}+(1-\frac{\phi_{\text{in}}^2}{4})^t\sqrt{\sum_{j=h+1}^{n}\mathbf{v}_j(v)^2}+\|\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\mathbf{v}_j-\sum_{j=1}^{h}\tilde{\mathbf{r}}_j(v)\cdot\tilde{\mathbf{r}}_j\|_2\\
&\leq (1-(1-\frac{\lambda_h}{2})^t)\sqrt{\sum_{j=1}^{h}\mathbf{v}_j(v)^2}+(1-\frac{\phi_{\text{in}}^2}{4})^t\sqrt{\sum_{j=h+1}^{n}\mathbf{v}_j(v)^2}+\|\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\mathbf{v}_j-\sum_{j=1}^{h}\tilde{\mathbf{r}}_j(v)\cdot\tilde{\mathbf{r}}_j\|_2\\
&\leq (1-(1-\phi_{\text{out}})^t)\cdot\sqrt{X_v}+(1-\frac{\phi_{\text{in}}^2}{4})^t+\|\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\mathbf{v}_j-\sum_{j=1}^{h}\tilde{\mathbf{r}}_j(v)\cdot\tilde{\mathbf{r}}_j\|_2\\
&\qquad\qquad\text{(by Lemma 5.2 and Fact 5.3)}\\
&\leq t\phi_{\text{out}}\cdot\sqrt{X_v}+\frac{1}{n^3}+\|\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\mathbf{v}_j-\sum_{j=1}^{h}\tilde{\mathbf{r}}_j(v)\cdot\tilde{\mathbf{r}}_j\|_2\quad(\text{by our setting } t=\frac{20\log n}{\phi_{\text{in}}^2})
\end{aligned}
$$

Now observe that

$$
\begin{aligned}
&\|\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\mathbf{v}_j-\sum_{j=1}^{h}\tilde{\mathbf{r}}_j(v)\cdot\tilde{\mathbf{r}}_j\|_2\\
=\;&\|\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\mathbf{v}_j-\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\tilde{\mathbf{r}}_j+\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\tilde{\mathbf{r}}_j-\sum_{j=1}^{h}\tilde{\mathbf{r}}_j(v)\cdot\tilde{\mathbf{r}}_j\|_2\\
\leq\;&\|\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\mathbf{v}_j-\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\tilde{\mathbf{r}}_j\|_2+\|\sum_{j=1}^{h}\mathbf{v}_j(v)\cdot\tilde{\mathbf{r}}_j-\sum_{j=1}^{h}\tilde{\mathbf{r}}_j(v)\cdot\tilde{\mathbf{r}}_j\|_2\\
\leq\;&\sum_{j=1}^{h}\|\mathbf{v}_j(v)\cdot(\mathbf{v}_j-\tilde{\mathbf{r}}_j)\|_2+\|\sum_{j=1}^{h}(\mathbf{v}_j(v)-\tilde{\mathbf{r}}_j(v))\cdot\tilde{\mathbf{r}}_j\|_2\\
\leq\;&\sqrt{c_{5.4}\cdot\frac{h\phi_{\text{out}}}{\phi_{\text{in}}^2}}\cdot\sum_{j=1}^{h}|\mathbf{v}_j(v)|+\|\sum_{j=1}^{h}(\mathbf{v}_j(v)-\tilde{\mathbf{r}}_j(v))\cdot\tilde{\mathbf{r}}_j\|_2\quad(\text{by Theorem 5.4})\\
\leq\;&\sqrt{c_{5.4}\cdot\frac{h^2\phi_{\text{out}}}{\phi_{\text{in}}^2}}\cdot\sqrt{\sum_{j=1}^{h}\mathbf{v}_j(v)^2}+\sqrt{\sum_{j=1}^{h}(\mathbf{v}_j(v)-\tilde{\mathbf{r}}_j(v))^2}\quad(\text{by Cauchy-Schwarz inequality})\\
=\;&\sqrt{c_{5.4}\cdot\frac{h^2\phi_{\text{out}}}{\phi_{\text{in}}^2}}\cdot\sqrt{X_v}+\sqrt{Y_v}
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\|\mathbf{p}_v^t-\mathcal{U}_{C_i}\|_2 &\leq t\phi_{\text{out}}\cdot\sqrt{X_v}+\frac{1}{n^3}+\sqrt{c_{5.4}\cdot\frac{h^2\phi_{\text{out}}}{\phi_{\text{in}}^2}}\cdot\sqrt{X_v}+\sqrt{Y_v}\\
&\leq \left(t\phi_{\text{out}}+\sqrt{c_{5.4}\cdot\frac{h^2\phi_{\text{out}}}{\phi_{\text{in}}^2}}\right)\cdot\sqrt{\frac{2h}{\beta\alpha n}}+\sqrt{c_{5.4}\cdot\frac{h^2\phi_{\text{out}}}{\phi_{\text{in}}^2}\frac{2}{\beta\alpha n}}+\frac{1}{n^3}\leq\frac{2\xi}{\sqrt{n}},
\end{aligned}
$$

where the last inequality follows from our setting that $h\leq k, t=\frac{20\log n}{\phi_{\text{in}}^2}$ and $\phi_{\text{out}}\leq a_{5.1}\frac{\xi\alpha\beta\phi_{\text{in}}^2}{k^3\log n}$, where $a_{5.1}>0$ is some sufficiently small constant.

Therefore, it holds that $\|\mathbf{p}_v^t-\mathcal{U}_{C_i}\|_{\text{TV}}=\frac{1}{2}\|\mathbf{p}_v^t-\mathcal{U}_{C_i}\|_1\leq\frac{1}{2}\sqrt{n}\cdot\|\mathbf{p}_v^t-\mathcal{U}_{C_i}\|_2\leq\xi$. ∎

5.2 From Clusterable Graphs to Noisy Clusterable Graphs

Now we analyze the random walk on a noisy clusterable graph G, for which we use an induced Markov chain introduced in [KPS13] and some property of stopping rules of Markov chains [LW97].

A tool: stopping rules of Markov Chains. Consider a finite, irreducible, discrete time Markov chain on the state space $V = [n]$ with stationary distribution π. For any distribution σ, we let σ^t denote the distribution of a t-step walk on the Markov chain with initial distribution σ. A *stopping rule* Γ of the Markov chain is a rule that observes the walk and decides whether to stop or not on the basis of what has been observed so far (see e.g., [LW97] for formal definition). Given a starting distribution σ and a target distribution τ, we say that a stopping rule Γ is a stopping rule from σ to τ if the initial state is drawn from σ and the final state is governed by τ. Let $\mathrm{E}[\Gamma]$ denote the expected length before Γ halts. For any two distributions σ and τ, we let $\mathcal{H}(\sigma, \tau)$ denote the minimal expected length $\mathrm{E}[\Gamma]$ among all stopping rules Γ from σ to τ.

Let $\sigma^{(t)}$ denotes the distribution of a uniform average walk of length t with initial distribution σ. The following lemma was proved by Lovász and Winkler.

Lemma 5.5 ([LW97])**.** *For any distribution τ, and any subset $U \subset V$,*

$$\sum_{i \in U} \sigma^{(t)}(i) \leq \frac{1}{t}\mathcal{H}(\sigma, \tau) + \frac{1}{t}\sum_{i \in U}\sum_{m=0}^{t-1} \tau^m(i)$$

where τ^m denotes the probability vector of an m step random walk on the Markov chain with initial distribution τ.

We remark that the above inequality was not explicitly stated in [LW97], while the proof of Lemma 4.22 in [LW97] directly implies the above Lemma.

An induced Markov chain. Let $G = (V, E)$ be a d-bounded graph. Let $\mathcal{M}$ be the Markov chain corresponding to the (normal) random walks on the input graph G. For simplicity, we assume $\mathcal{M}$ is irreducible (i.e., the graph is connected). By definition, the stationary distribution π of $\mathcal{M}$ is the uniform distribution $\mathcal{U}_V$ on V, that is $\pi(i) = \frac{1}{n}$. Let D denote a (large) subset of V and let $B = V \setminus D$. Now we describe the new Markov chain $\mathcal{M}'$, that has been considered in [KPS13], with state set D as follows. For any two vertices $u, v \in D$, the transition probability $\mathbf{p}'_u(v)$ in $\mathcal{M}'$ is the sum of $\mathbf{p}_u(v)$, i.e., the transition probability from u to v in $\mathcal{M}$, and the probability $\mathbf{b}_u^{(t)}(v)$ that is equal to the total probability of all length t walks from u to v all of whose states, except for the end points u and v are in B, for any integer $t \geq 2$. That is, $\mathbf{p}'_u(v) = \mathbf{p}_u(v) + \sum_{t \geq 2} \mathbf{b}_u^{(t)}(v)$. The chain $\mathcal{M}'$ is formally constructed by first retaining the original transition in $\mathcal{M}$ between u, v and then adding new transitions $e_u^{(t)}(v)$ with transition probability $\mathbf{b}_u^{(t)}(v)$ for any $t \geq 2$, for any $u, v \in D$.

We note that the chain $\mathcal{M}'$ is the *stochastic complement* of $\mathcal{M}$ with respect to set D [Mey89]. Let $\mathbf{P} = \left(\begin{smallmatrix} \mathbf{P}_D & \mathbf{P}_1 \\ \mathbf{P}_2 & \mathbf{P}_B \end{smallmatrix}\right)$ denote the transition probability matrix underlying $\mathcal{M}$. We have the following lemma regarding the transition probability matrix $\mathbf{P}'$ underlying $\mathcal{M}'$.

Lemma 5.6 ([Mey89])**.** *The Markov chain $\mathcal{M}'$ is irreducible and aperiodic. Furthermore, its transition probability matrix is $\mathbf{P}' = \mathbf{P}_D + \mathbf{P}_1(\mathbf{I} - \mathbf{P}_B)^{-1}\mathbf{P}_2$.*

It is known (see e.g., [Mey89] and [KPS13]) that, the stationary distribution in $\mathcal{M}'$ is given by the vector $\pi' \in \mathbb{R}^D$ such that $\pi'(u) = \frac{\pi(u)}{\pi(D)} = \frac{1}{|D|}$ for any $u \in D$.

Now let us consider a vertex $s \in D$ and an integer ℓ that will be specified later. Let $\tau := \mathbf{p}'^{(\ell)}_s$ denote the distribution of a random walk of length ℓ starting from $s \in D$ in $\mathcal{M}'$. Consider the stopping rule Γ that stops the walk in $\mathcal{M}$ as soon as it has taken ℓ steps in $\mathcal{M}'$, that is, Γ is a stopping rule from $\mathbf{1}_s$ to τ. Recall

that $\mathrm{E}[\Gamma]$ denotes the expected number of steps the walk takes starting from s before being terminated by the stopping rule Γ. The following lemma has been proven in [KPS13].

Lemma 5.7 ([KPS13])**.** *There exists a set $\tilde{B} \subseteq D$ with $\pi(\tilde{B}) \leq \pi(B)$ such that for any $s \in D \setminus \tilde{B}$, $E[\Gamma] \leq 2\ell$. In particular, for any such vertex s, $\mathcal{H}(\boldsymbol{1}_s, \tau) \leq 2\ell$.*

Now we use the above induced chain to analyze the random walks on noisy clusterable graphs. Let G be a graph with an h-partition C_i, $i \leq h$ satisfying the precondition of Theorem 1.5. We let D denote the union of all D_i's with $|D_i| \geq 2|B_i|$, that is, $D = \cup_{i:|D_i|\geq 2|B_i|} D_i$ and $B = V \setminus D$. We consider the induced Markov chain $\mathcal{M}'$ with state set D.

Recall that we let $\mathbf{A}$ denote the adjacency matrix of the d-regular graph G' corresponding to G (see Section 2.) Then the transition probability matrix is $\mathbf{P} = \frac{\mathbf{I}+\frac{1}{d}\mathbf{A}}{2}$. If we let $\mathbf{A} = \left(\begin{smallmatrix} \mathbf{A}_D & \mathbf{A}_1 \\ \mathbf{A}_2 & \mathbf{A}_B \end{smallmatrix}\right)$, then by Lemma 5.6, the transition probability matrix of $G_{\mathbf{M}'}$ is

$$\mathbf{P}' = \frac{\mathbf{I} + \frac{1}{d}\mathbf{A}_D}{2} + \frac{\mathbf{A}_1}{2d}\left(\frac{\mathbf{I} - \frac{1}{d}\mathbf{A}_B}{2}\right)^{-1}\frac{\mathbf{A}_2}{2d} = \frac{\mathbf{I} + \frac{1}{d}(\mathbf{A}_D + \mathbf{A}_1(2d\mathbf{I} - \mathbf{A}_B)^{-1}\mathbf{A}_2)}{2}. \tag{1}$$

If we let $G_{\mathcal{M}'}$ denote the (weighted) d-bounded graph with adjacency matrix $\mathbf{A}_D + \mathbf{A}_1(2d\mathbf{I} - \mathbf{A}_B)^{-1}\mathbf{A}_2$, then by the above analysis (and the fact that $(2d\mathbf{I} - \mathbf{A}_B)^{-1} \geq \mathbf{0}$ [Mey89]), $\mathcal{M}'$ corresponds to the lazy random walk on the graph $G_{\mathcal{M}'}$.

In the following, we show that $G_{\mathcal{M}'}$ is a clusterable graph with clusters $D_i \subseteq D$, which will imply that the chain $\mathcal{M}'$ has the nice local mixing property as guaranteed by Theorem 5.1. Then we can use the stopping rules to relate the chains $\mathcal{M}'$ and $\mathcal{M}$.

The following lemma shows that if we construct $\mathcal{M}'$ as above for the graph that satisfies the precondition of Theorem 1.5, then $G_{\mathcal{M}'}$ is $(k, \phi_{\text{in}}, O(\phi_{\text{out}}))$-clusterable. This is trivial for the case of $k = 1$ (as in [KPS13]), as the inner conductance of any set is monotonically increasing. However, for general $k \geq 2$, we need to deal with the difficulty of bounding the outer conductance of potential clusters, as the outer conductance of any set is also monotonically increasing due to our construction.

Lemma 5.8. *Let $G = (V, E)$ be a d-bound graph with an h-partition C_i, $i \leq h$ such that $\phi_G(C_i) \leq \phi_{out}$. Furthermore, each C_i can be partitioned into two subsets D_i and B_i such that $\phi(G[D_i]) \geq \phi_{in}$. Let $D = \cup_{i:|D_i|\geq 2|B_i|} D_i$ and $B = V \setminus D$. Let $G_{\mathcal{M}'}$ be the weighted graph corresponding to the Markov chain $\mathcal{M}'$ on D constructed as above. Then in the graph $G_{\mathcal{M}'}$, each $D_i \subseteq D$ has the inner conductance at least ϕ_{in} and outer conductance at most $3\phi_{out}$.*

Proof. We first consider the inner conductance of D_i in $G_{\mathcal{M}'}$. Let $S \subseteq D_i$ with $|S| \leq \frac{|D_i|}{2}$. By the fact that the adjacency matrix of $G_{\mathcal{M}'}$ is $\mathbf{A}_D + \mathbf{A}_1(2d\mathbf{I} - \mathbf{A}_B)^{-1}\mathbf{A}_2$, it holds that $|E_{G_{\mathcal{M}'}}(S, D_i \setminus S)| \geq |E_G(S, D_i \setminus S)| \geq \phi_{\text{in}} d|S|$. This implies that the inner conductance of D_i in $G_{\mathcal{M}'}$ is at least ϕ_{in}.

To bound the outer conductance of D_i in $G_{\mathcal{M}'}$, we instead bound the outer conductance $\phi_{\mathcal{M}'}(D_i)$ of D_i in the Markov chain $\mathcal{M}'$, which is defined to be $\phi_{\mathcal{M}'}(D_i) := \frac{\sum_{u\in D_i, v\in D\setminus D_i} \pi'(u)\mathbf{p}'_u(v)}{\pi'(D_i)}$, where $\mathbf{p}'_u(v)$ denotes the transition probability from u to v in the Markov chain $\mathcal{M}'$. Note that by our definitions, $\phi_{G_{\mathcal{M}'}}(D_i) = 2\phi_{\mathcal{M}'}(D_i)$.

Recall that $\pi'(u) = \frac{1}{|D|}$ and that the transition probability matrix of $\mathcal{M}'$ is $\mathbf{P}'$ given by Equation (1). Then we have that

$$\sum_{u\in D_i, v\in D\setminus D_i} \pi'(u)\mathbf{p}'_u(v) = \frac{1}{|D|}\sum_{u\in D_i, v\in D\setminus D_i} \mathbf{1}_u \cdot \mathbf{P}' \cdot \mathbf{1}_v^T$$

$$= \frac{1}{|D|}\sum_{u\in D_i, v\in D\setminus D_i} \mathbf{1}_u \cdot \left(\frac{\mathbf{I} + \frac{1}{d}(\mathbf{A}_D + \mathbf{A}_1(2d\mathbf{I} - \mathbf{A}_B)^{-1}\mathbf{A}_2)}{2}\right) \cdot \mathbf{1}_v^T$$

$$
\begin{aligned}
&= \frac{1}{|D|}\sum_{u\in D_i, v\in D\setminus D_i} \mathbf{1}_u \cdot \left(\frac{1}{2d}\left(\mathbf{A}_D + \mathbf{A}_1(2d\mathbf{I}-\mathbf{A}_B)^{-1}\mathbf{A}_2\right)\right)\cdot \mathbf{1}_v^T \\
&= \frac{1}{|D|}\sum_{u\in D_i, v\in D\setminus D_i} \left(\frac{1}{2d}\left(\mathbf{1}_u\cdot\mathbf{A}_D\cdot\mathbf{1}_v^T + \frac{1}{2d}\mathbf{1}_u\cdot\mathbf{A}_1\cdot\sum_{j=0}^{\infty}(\frac{1}{2d}\mathbf{A}_B)^j\cdot\mathbf{A}_2\cdot\mathbf{1}_v^T\right)\right)
\end{aligned} \tag{2}
$$

where last equation follows from the Neumann Series $(\mathbf{I}-\frac{\mathbf{A}_B}{2d})^{-1} = \sum_{j=0}^{\infty}(\frac{\mathbf{A}_B}{2d})^j$.

We bound each term in the right hand side of the above inequality as follows. First, we have that

$$
\sum_{u\in D_i, v\in D\setminus D_i} \mathbf{1}_u\cdot\mathbf{A}_D\cdot\mathbf{1}_v^T \le |E_G(D_i, D\setminus D_i)|. \tag{3}
$$

Furthermore, we observe that $\mathbf{1}_u\cdot\mathbf{A}_1\cdot\mathbf{A}_2\cdot\mathbf{1}_v^T$ is exactly the number of paths that start from u, then go to a vertex $w\in B$, and then move to v. Thus,

$$
\begin{aligned}
\frac{1}{2d}\sum_{u\in D_i, v\in D\setminus D_i}\mathbf{1}_u\cdot\mathbf{A}_1\mathbf{A}_2\cdot\mathbf{1}_v^T &\le \sum_{w\in B}\frac{|E_G(D_i,w)||E_G(w,D\setminus D_i)|}{2d} \\
&\le \sum_{w\in B_i}\frac{|E_G(w,D\setminus D_i)|}{2} + \sum_{w\in B\setminus B_i}\frac{|E_G(D_i,w)|}{2} \\
&= \frac{1}{2}(|E_G(B_i,D\setminus D_i)| + |E_G(D_i,B\setminus B_i)|) \\
&\le \frac{1}{2}|E_G(C_i, V\setminus C_i)|
\end{aligned}
$$

Similarly, for each $j\ge 1$, $\mathbf{1}_u\cdot\mathbf{A}_1\cdot\mathbf{A}_B^j\cdot\mathbf{A}_2\cdot\mathbf{1}_v^T$ is exactly the number of paths that start from u, then go to a vertex $w_1\in B$, and move inside B for the next j steps until some vertex $w_2\in B$, and then move to v. We have that

$$
\begin{aligned}
&\frac{1}{(2d)^{j+1}}\sum_{u\in D_i, v\in D\setminus D_i}\mathbf{1}_u\cdot\mathbf{A}_1\mathbf{A}_B^j\mathbf{A}_2\cdot\mathbf{1}_v^T \\
\le\ & \frac{1}{(2d)^{j+1}}\sum_{w_1\in B}|E_G(D_i,w_1)|\cdot\sum_{\substack{w_2: p=(v_0=w_1,\cdots,v_j=w_2),\\ v_\ell\in B,(v_\ell,v_{\ell+1})\in E(G)}}|E_G(w_2,D\setminus D_i)| \\
\le\ & \frac{1}{(2d)^{j+1}}(\sum_{w_1\in B\setminus B_i}|E_G(D_i,w_1)|\cdot d^{j+1} + \sum_{w_2\in B_i}|E_G(w_2,D\setminus D_i)|\cdot d^{j+1}) \\
=\ & \frac{1}{2^{j+1}}(|E_G(D_i,B\setminus B_i)| + |E_G(B_i,D\setminus D_i)|) \\
\le\ & \frac{1}{2^{j+1}}|E_G(C_i,V\setminus C_i)|,
\end{aligned}
$$

where in the first inequality, the third summation is taken over all possible paths p from w_1 to some vertex $w_2\in B$, such that the length of p is j and all vertices on p belong to B; in the second inequality, we used the fact that the number of such paths p is at most d^j and each vertex has degree at most d.

Thus,

$$
\sum_{j=0}^{\infty}\frac{1}{(2d)^{j+1}}\sum_{u\in D_i, v\in D\setminus D_i}\mathbf{1}_u\cdot\mathbf{A}_1\mathbf{A}_B^i\mathbf{A}_2\cdot\mathbf{1}_v^T \le \sum_{j=0}^{\infty}\frac{1}{2^{j+1}}|E_G(C_i,V\setminus C_i)| = |E_G(C_i,V\setminus C_i)| \tag{4}
$$

By the above inequalities (2),(3),(4), we obtain that

$$\begin{aligned}\sum_{u\in D_i, v\in D\setminus D_i} \pi'(u)\mathbf{p}'_u(v) &\leq \frac{1}{2d|D|}\cdot(1+1)\cdot|E_G(C_i, V\setminus C_i)| = \frac{|E_G(C_i, V\setminus C_i)|}{d|C_i|}\cdot\frac{|C_i|}{|D_i|}\cdot\frac{|D_i|}{|D|}\\ &\leq \phi_G(C_i)\cdot\frac{3}{2}\cdot\pi'(D_i) \leq \frac{3}{2}\phi_{\text{out}}\pi'(D_i),\end{aligned}$$

where in the second to last inequality, we used the assumption that $|D_i| \geq 2|B_i|$, which gives that $|D_i| \geq \frac{2}{3}|C_i|$.

Therefore, $\phi_{G_{\mathcal{M}'}}(D_i) = 2\phi_{\mathcal{M}'}(D_i) \leq 3\phi_{\text{out}}$.

Now we are ready to prove Theorem 1.5.

Proof of Theorem 1.5. Let $D = \cup_{j:|D_j|\geq 2|B_j|} D_j$. Let $B = V\setminus D$. Then it holds that $|B| = \sum_{1\leq i\leq h}|B_i| + \sum_{i:|D_i|<2|B_i|}|D_i| \leq 3\sum_{1\leq i\leq h}|B_i| \leq 3\varepsilon n$, and $|D| \geq (1-3\varepsilon)n$. We consider the induced Markov chain $\mathcal{M}'$ on D as above. By Lemma 5.8, the corresponding d-bounded weighted graph $G_{\mathcal{M}'}$ is $(k, \phi_{\text{in}}, 3\phi_{\text{out}})$-clusterable. In particular, $\phi_{G_{\mathcal{M}'}}(D_i) \leq 3\phi_{\text{out}}$ and $\phi(G_{\mathcal{M}'}[D_i]) \geq \phi_{\text{in}}$ for any $D_i \subset D$.

Let ℓ be an integer that will be specified later. For any $s \in D$, we let $\tau_s := \mathbf{p}'_s{}^{(\ell)}$ being the probability distribution of an ℓ step random walk starting from s in the induced Markov chain $\mathcal{M}'$. Let Γ_s be the stopping rule from $\mathbf{1}_s$ to τ_s which is obtained by stopping the random walk that starts at s in $\mathcal{M}$ as soon as it has taken ℓ steps in $\mathcal{M}'$. Let $\tilde{B} \subseteq D$ be the set guaranteed by Lemma 5.7 such that $|\tilde{B}| \leq |B| \leq 3\varepsilon n$ and for any $s \in D\setminus\tilde{B}$,

$$E[\Gamma_s] \leq 2\ell. \tag{5}$$

Now we set $a_{1.5} = \frac{a_{5.1}}{120}$ and thus $\phi_{\text{out}} \leq \frac{a_{5.1}\varepsilon\gamma^4\phi_{\text{in}}^2}{120k^3\log n}$. We then apply Theorem 5.1 on $G_{\mathcal{M}'}$ with $(\phi_{\text{in}}, 3\phi_{\text{out}})$-clusters D_i and $\alpha = 3\sqrt{\varepsilon}, \beta = 3\sqrt{\varepsilon}$, $\xi = \frac{\gamma}{6}$, to obtain that for any D_j with $|D_j| \geq 3\sqrt{\varepsilon}n \geq 3\sqrt{\varepsilon}|D|$,there exists a set D'_j with $|D'_j| \geq (1-3\sqrt{\varepsilon})|D_j|$ such that for any $s \in D'_j$ and $\ell = \frac{20\log n}{\phi_{\text{in}}^2}$, it holds that $\|\tau_s - \mathcal{U}_{D_j}\|_{\text{TV}} \leq \frac{\gamma}{6}$. This implies that

$$\begin{aligned}\|\tau_s - \mathcal{U}_{C_j}\|_{\text{TV}} &\leq \|\tau_s - \mathcal{U}_{D_j}\|_{\text{TV}} + \|\mathcal{U}_{D_j} - \mathcal{U}_{C_j}\|_{\text{TV}} \leq \frac{\gamma}{6} + \frac{|C_j\setminus D_j|}{|C_j|}\\ &= \frac{\gamma}{6} + \frac{|B_j|}{|C_j|} \leq \frac{\gamma}{6} + \frac{\varepsilon n}{3\sqrt{\varepsilon}n} = \frac{\gamma}{6} + \frac{\sqrt{\varepsilon}}{3}\end{aligned} \tag{6}$$

Now we set $\widehat{D}_j := D'_j\setminus\tilde{B}$. Then it is guaranteed that for any j with $|D_j| \geq 3\sqrt{\varepsilon}n$, $|\widehat{D}_j| \geq (1-3\sqrt{\varepsilon})|D_j| - 3\varepsilon n \geq (1-4\sqrt{\varepsilon})|D_j|$. Thus, for any $s \in \widehat{D}_j$, both inequalities (5) and (6) hold.

Now let us consider an arbitrary $s \in \widehat{D}_j$. Let $\tau = \tau_s$ and $\sigma = \mathbf{1}_s$. By the precondition of the Theorem, we have that $t = \frac{120\log n}{\gamma\phi_{\text{in}}^2} = \frac{6\ell}{\gamma}$. We further recall that $\mathbf{a}_s^t$ denotes the distribution of a uniform average walk of length t with initial distribution σ in the original chain $\mathcal{M}$. By applying Lemma 5.5 with $\sigma^{(t)} = \mathbf{a}_s^t$ and distribution τ, we obtain that for any $U \subset V$,

$$\sum_{i\in U}\mathbf{a}_s^t(i) \leq \frac{1}{t}\mathcal{H}(\sigma,\tau) + \frac{1}{t}\sum_{i\in U}\sum_{m=0}^{t-1}\tau^m(i),$$

where τ^m denotes the distribution of an m step random walk on G with initial distribution τ, that is $\tau^m = \tau\mathbf{P}^m$. (Here we slightly abuse the notation τ and use it to denote the distribution on V by adding zero

coordinates corresponding to vertices in $V \setminus D$). This further implies that for any set C_j and any $U \subseteq V$,

$$\sum_{i\in U}(\mathbf{a}_s^t(i) - \mathcal{U}_{C_j}(i)) \le \frac{1}{t}\mathcal{H}(\sigma,\tau) + \frac{1}{t}\sum_{i\in U}\sum_{m=0}^{t-1}(\tau^m(i) - \mathcal{U}_{C_j}(i))$$

Therefore,

$$\|\mathbf{a}_s^t - \mathcal{U}_{C_j}\|_{\text{TV}} \;\le\; \frac{1}{t}\mathcal{H}(\sigma,\tau) + \frac{1}{t}\sum_{m=0}^{t-1}\|\tau^m - \mathcal{U}_{C_j}\|_{\text{TV}} \le \frac{2\ell}{t} + \frac{1}{t}\sum_{m=0}^{t-1}\|\tau^m - \mathcal{U}_{C_j}\|_{\text{TV}}, \tag{7}$$

where the last inequality follows from inequality (5). Now recall that $\mathbf{P} = \frac{\mathbf{I}+d^{-1}\mathbf{A}}{2}$ denotes the transition probability matrix of the random walk. We will show the following claim.

Claim 5.9. *For any $0 \le m \le t-1$, it holds that $\|\mathcal{U}_{C_j}\mathbf{P}^m - \mathcal{U}_{C_j}\|_{TV} \le \frac{\gamma}{3}$.*

Assuming that the above claim holds, we have that for any $0 \le m \le t-1$,

$$\begin{aligned}
&\|\tau^m - \mathcal{U}_{C_j}\|_{\text{TV}} = \|\tau\mathbf{P}^m - \mathcal{U}_{C_j}\|_{\text{TV}} \le \|\tau\mathbf{P}^m - \mathcal{U}_{C_j}\mathbf{P}^m + \mathcal{U}_{C_j}\mathbf{P}^m - \mathcal{U}_{C_j}\|_{\text{TV}}\\
\le\;& \|\tau\mathbf{P}^m - \mathcal{U}_{C_j}\mathbf{P}^m\|_{\text{TV}} + \|\mathcal{U}_{C_j}\mathbf{P}^m - \mathcal{U}_{C_j}\|_{\text{TV}} \le \|\tau - \mathcal{U}_{C_j}\|_{\text{TV}} + \|\mathcal{U}_{C_j}\mathbf{P}^m - \mathcal{U}_{C_j}\|_{\text{TV}}\\
\le\;& \frac{\gamma}{6} + \frac{\sqrt{\varepsilon}}{3} + \frac{\gamma}{3} = \frac{\gamma}{2} + \frac{\sqrt{\varepsilon}}{3},
\end{aligned}$$

where the last inequality follows from Ineq. (6) and Claim 5.9. This, together with inequality (7), gives that

$$\|\mathbf{a}_s^t - \mathcal{U}_{C_j}\|_{\text{TV}} \le \frac{2\ell}{t} + \frac{1}{t}\cdot t\cdot(\frac{\gamma}{2} + \frac{\sqrt{\varepsilon}}{3}) \le \frac{\gamma}{3} + \frac{\gamma}{2} + \frac{\sqrt{\varepsilon}}{3} < \gamma + \sqrt{\varepsilon}.$$

This will then finish the proof of the theorem.

Now we give the proof of Claim 5.9.

Proof of Claim 5.9. For notational simplicity, we let $C = C_j$. We write $\mathbf{P} = \sum_{i=1}^n \eta_i \mathbf{v}_i\mathbf{v}_i^T$, where $\eta_i := 1 - \frac{\lambda_i}{2}$ and $\mathbf{v}_i$ ($1 \le i \le n$) denote the i-th eigenvalue of $\mathbf{P}$, respectively. Let $\mathcal{U}_C = \sum_i \alpha_i\mathbf{v}_i$. Note that $\sum_{i=1}^n \alpha_i^2 = \|\mathcal{U}_C\|_2^2 = \frac{1}{|C|}$.

Note that

$$\frac{\mathbf{1}_C}{|C|}\cdot(\mathbf{I}-\mathbf{P})\mathbf{1}_C^T = \frac{\mathbf{1}_C(d\mathbf{I}-\mathbf{A})\mathbf{1}_C^T}{2d|C|} = \frac{\sum_{u\sim v}(\mathbf{1}_C(u) - \mathbf{1}_C(v))^2}{2d|C|} = \frac{\phi_G(C)}{2} \le \frac{\phi_{\text{out}}}{2},$$

which gives that $1 - |C|\cdot\mathcal{U}_C\mathbf{P}\mathcal{U}_C^T \le \frac{\phi_{\text{out}}}{2}$. Thus, $1 - |C|\sum_i \eta_i\alpha_i^2 \le \frac{\phi_{\text{out}}}{2}$, or equivalently, $\sum_i \eta_i\alpha_i^2 \ge \frac{1-\phi_{\text{out}}/2}{|C|}$.

Let $H = \{i : \eta_i \ge 1 - \frac{x\phi_{\text{out}}}{2}\}$, where $x = \frac{8}{\gamma^2}$. Then we have that $\sum_{i\in H}\alpha_i^2 + (1-\frac{x\phi_{\text{out}}}{2})\sum_{i\notin H}\alpha_i^2 \ge \frac{1-\phi_{\text{out}}/2}{|C|}$. Thus, $\sum_{i\in H}\alpha_i^2 + (1-\frac{x\phi_{\text{out}}}{2})(\frac{1}{|C|} - \sum_{i\in H}\alpha_i^2) \ge \frac{1-\phi_{\text{out}}/2}{|C|}$, which gives that

$$\sum_{i\in H}\alpha_i^2 \ge \frac{x-1}{x\cdot|C|}, \quad \sum_{i\notin H}\alpha_i^2 \le \frac{1}{x|C|}.$$

Now we have that

$$\begin{aligned}
&\|\mathcal{U}_C\mathbf{P}^m - \mathcal{U}_C\|_2^2 = \sum_i(\alpha_i\eta_i^m - \alpha_i)^2 = \sum_i \alpha_i^2(1-\eta_i^m)^2 \le \sum_{i\in H}(1-(1-\frac{x\phi_{\text{out}}}{2})^m)^2\alpha_i^2 + \sum_{i\notin H}\alpha_i^2\\
\le\;& \sum_{i\in H}(\frac{xt\phi_{\text{out}}}{2})^2\alpha_i^2 + \frac{1}{x|C|} \le (\frac{x^2t^2\phi_{\text{out}}^2}{4} + \frac{1}{x})\frac{1}{|C|} < \frac{\gamma^2}{4|C|},
\end{aligned}$$

where we used our choice of parameters which satisfy that $t\phi_{\text{out}} \leq \gamma^3/16$ and $x = \frac{8}{\gamma^2}$.

On the other hand, if we let $\mathbf{D}_C$ denote the diagonal matrix such that $\mathbf{D}_C(u,u) = 1$ if $u \in C$ and 0 otherwise, then by Proposition 2.5 in [ST13], it holds that for any $m \geq 0$,

$$\mathcal{U}_C(\mathbf{P}\mathbf{D}_C)^m\mathbf{1}_C^T = \mathcal{U}_C(\mathbf{P}\mathbf{D}_C)^m\mathbf{1}_V^T \geq 1 - \frac{m\phi_G(C)}{2} \geq 1 - \frac{m\phi_{\text{out}}}{2}.$$

This gives that

$$\mathcal{U}_C\mathbf{P}^m\mathbf{1}_{V\setminus C}^T = 1 - \mathcal{U}_C\mathbf{P}^m\mathbf{1}_C^T \leq 1 - \mathcal{U}_C(\mathbf{P}\mathbf{D}_C)^m\mathbf{1}_C^T \leq \frac{m\phi_{\text{out}}}{2}.$$

Finally, by the above calculations, we have that

$$\begin{aligned}
&\|\mathcal{U}_C\mathbf{P}^m - \mathcal{U}_C\|_{\text{TV}} = \frac{1}{2}\|\mathcal{U}_C\mathbf{P}^m - \mathcal{U}_C\|_1 \leq \frac{1}{2}(\mathcal{U}_C\mathbf{P}^m\mathbf{1}_{V\setminus C}^T + \sum_{i\in C}|\mathcal{U}_C\mathbf{P}^m(i) - \mathcal{U}_C(i)|)\\
\leq\ & \frac{1}{2}(\frac{m\phi_{\text{out}}}{2} + \sqrt{|C|}\cdot\sqrt{\sum_{i\in C}(\mathcal{U}_C\mathbf{P}^m(i) - \mathcal{U}_C(i))^2}) \leq \frac{1}{2}(\frac{t\phi_{\text{out}}}{2} + \sqrt{|C|}\cdot\|\mathcal{U}_C\mathbf{P}^m - \mathcal{U}_C\|_2)\\
\leq\ & \frac{\gamma^3}{64} + \frac{\gamma}{4} < \frac{\gamma}{3}.
\end{aligned}$$

This finishes the proof of the Claim. ∎

This finishes the proof of Theorem 1.5. ∎

6 Conclusions

We gave the first robust clustering oracle and local filter for reconstructing the cluster structure of bounded degree graphs. Both algorithms run in sublinear times. To design and analyze our algorithms, we formalized and proved a new behavior of random walks in a noisy clusterable graph: a random walk of appropriately chosen length from a typical vertex in a large cluster of the clusterable part will mix well in the corresponding cluster, which might be of independent interest.

It will be an interesting open question to design a local reconstruction algorithm that outputs a clusterable graph with better cluster-quality guarantee, especially to remove the $\Theta(\log n)$ gap between the inner conductances of the original graph and the corrected graph from our current result. In the property testing setting, such a gap was successfully closed, for both testing expansion ([CS10] vs. [KS11, NS10]) and for testing k-clusterability ([CPS15] vs. [CKK$^+$18]). However, for the local reconstruction setting, we even do not know how to remove such a logarithmic gap for reconstructing noisy expander graphs (i.e., $k = 1$). As noted in [KPS13], for the case $k = 1$, one already needs to have more refined definitions of strong/weak vertices and much stronger results about random walks in noisy expander graphs. Removing the logarithmic gap from our result for locally reconstructing cluster structure for general $k \geq 1$ can be as hard, if not harder. Similar question can be asked for removing the $\Theta(\log n)$ gap between the inner and outer conductance of the input instance of our robust clustering oracle. As we mentioned before, there is evidence in [CKK$^+$18] showing that this is difficult (for distribution distance based algorithms).

Acknowledgements. We are thankful to anonymous reviewers of FOCS 2018 and STOC 2019 for valuable comments.

References

[ABJ18] Nir Ailon, Anup Bhattacharya, and Ragesh Jaiswal. Approximate correlation clustering using same-cluster queries. In *Latin American Symposium on Theoretical Informatics*, pages 14–27. Springer, 2018.

[ABJK18] Nir Ailon, Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Approximate clustering with same-cluster queries. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 40:1–40:21, 2018.

[ACCL08] Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Property-preserving data reconstruction. *Algorithmica*, 51(2):160–182, 2008.

[ACL06] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 475–486. IEEE, 2006.

[AKBD16] Hassan Ashtiani, Shrinu Kushagra, and Shai Ben-David. Clustering with same-cluster queries. In *Advances in neural information processing systems*, pages 3216–3224, 2016.

[AOPT16] Reid Andersen, Shayan Oveis Gharan, Yuval Peres, and Luca Trevisan. Almost optimal local graph clustering using evolving sets. *Journal of the ACM (JACM)*, 63(2):15, 2016.

[AP09] Reid Andersen and Yuval Peres. Finding sparse cuts locally using evolving sets. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 235–244. ACM, 2009.

[ARVX12] Noga Alon, Ronitt Rubinfeld, Shai Vardi, and Ning Xie. Space-efficient local computation algorithms. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1132–1139. Society for Industrial and Applied Mathematics, 2012.

[AT10] Tim Austin and Terence Tao. Testability and repair of hereditary hypergraph properties. *Random Structures & Algorithms*, 36(4):373–463, 2010.

[Bra08] Zvika Brakerski. Local property restoring. *Unpublished manuscript*, 2008.

[BSS10] Itai Benjamini, Oded Schramm, and Asaf Shapira. Every minor-closed property of sparse graphs is testable. *Advances in Mathematics*, 6(223):2200–2218, 2010.

[CGR13] Andrea Campagna, Alan Guo, and Ronitt Rubinfeld. Local reconstructors and tolerant testers for connectivity and diameter. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 411–424. Springer, 2013.

[CJSX14] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. *The Journal of Machine Learning Research*, 15(1):2213–2238, 2014.

[CKK+18] Ashish Chiplunkar, Michael Kapralov, Sanjeev Khanna, Aida Mousavifar, and Yuval Peres. Testing graph clusterability: Algorithms and lower bounds. In *59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2018.

[CL15] T Tony Cai and Xiaodong Li. Robust and computationally feasible community detection in the presence of arbitrary outlier nodes. *The Annals of Statistics*, 43(3):1027–1059, 2015.

[CPS15] Artur Czumaj, Pan Peng, and Christian Sohler. Testing cluster structure of graphs. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 723–732. ACM, 2015.

[CS10] Artur Czumaj and Christian Sohler. Testing expansion in bounded-degree graphs. *Combinatorics, Probability and Computing*, 19(5-6):693–709, 2010.

[CS11] Bernard Chazelle and C Seshadhri. Online geometric reconstruction. *Journal of the ACM (JACM)*, 58(4):14, 2011.

[DGK17] Roee David, Elazar Goldenberg, and Robert Krauthgamer. Local reconstruction of low-rank matrices and subspaces. *Random Structures & Algorithms*, 2017.

[DLRR13] Akashnil Dutta, Reut Levi, Dana Ron, and Ronitt Rubinfeld. A simple online competitive adaptation of lempel-ziv compression with efficient random access support. In *Data Compression Conference (DCC), 2013*, pages 113–122. IEEE, 2013.

[DPRS19] Tamal K Dey, Pan Peng, Alfred Rossi, and Anastasios Sidiropoulos. Spectral concentration and greedy k-clustering. *Computational Geometry*, 76:19–32, 2019.

[For10] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[GG81] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, 1981.

[GGR98] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

[GR98] Oded Goldreich and Dana Ron. A sublinear bipartiteness tester for bounded degree graphs. In *Proceedings of the thirtieth Annual ACM Symposium on Theory of Computing (STOC)*, pages 289–298. ACM, 1998.

[GR00] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(20), 2000.

[GR02] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002.

[GRSY14] Amir Globerson, Tim Roughgarden, David Sontag, and Cafer Yildirim. Tight error bounds for structured prediction. *arXiv preprint arXiv:1409.5834*, 2014.

[GV16] Olivier Guédon and Roman Vershynin. Community detection in sparse networks via grothendiecks inequality. *Probability Theory and Related Fields*, 165(3-4):1025–1049, 2016.

[HKNO09] Avinatan Hassidim, Jonathan A Kelner, Huy N Nguyen, and Krzysztof Onak. Local graph partitions for approximation and testing. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 22–31. IEEE, 2009.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[JR13] Madhav Jha and Sofya Raskhodnikova. Testing and reconstruction of lipschitz functions with applications to data privacy. *SIAM Journal on Computing*, 42(2):700–731, 2013.

[KPS13] Satyen Kale, Yuval Peres, and C Seshadhri. Noise tolerance of expanders and sublinear expansion reconstruction. *SIAM Journal on Computing*, 42(1):305–323, 2013.

[KS11] Satyen Kale and C. Seshadhri. An expansion tester for bounded degree graphs. *SIAM Journal on Computing*, 40(3):709–720, 2011.

[KSS18] Akash Kumar, C. Seshadhri, and Andrew Stolman. Finding forbidden minors in sublinear time: an $n^{1/2+o(1)}$-query one-sided tester for minor closed properties. In *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2018.

[KVV04] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.

[LW97] László Lovász and Peter Winkler. Mixing times. In *Microsurveys in Discrete Probability, Proceedings of a DIMACS Workshop, Princeton, New Jersey, USA, 1997*, pages 85–134, 1997.

[Mey89] Carl D Meyer. Stochastic complementation, uncoupling markov chains, and the theory of nearly reducible systems. *SIAM review*, 31(2):240–272, 1989.

[MMV12] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 367–384. ACM, 2012.

[MMV14] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Constant factor approximation for balanced cut in the pie model. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 41–49. ACM, 2014.

[MMV15] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Correlation clustering with noisy partial information. In *Proceedings of The 28th Conference on Learning Theory*, pages 1321–1342, 2015.

[MMV16] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Learning communities in the presence of errors. In *Conference on Learning Theory*, pages 1258–1291, 2016.

[MPW16] Ankur Moitra, William Perry, and Alexander S Wein. How robust are reconstruction thresholds for community detection? In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 828–841. ACM, 2016.

[MRVX12] Yishay Mansour, Aviad Rubinstein, Shai Vardi, and Ning Xie. Converting online algorithms to local computation algorithms. *Automata, Languages, and Programming*, pages 653–664, 2012.

[MS10] Claire Mathieu and Warren Schudy. Correlation clustering with noisy input. In *Proceedings of the twenty-first annual ACM-SIAM symposium on discrete algorithms*, pages 712–728. Society for Industrial and Applied Mathematics, 2010.

[MS17a] Arya Mazumdar and Barna Saha. Clustering with noisy queries. In *Advances in Neural Information Processing Systems*, pages 5788–5799, 2017.

[MS17b] Arya Mazumdar and Barna Saha. Query complexity of clustering with side information. In *Advances in Neural Information Processing Systems*, pages 4682–4693, 2017.

[MV13] Yishay Mansour and Shai Vardi. A local computation approximation scheme to maximum matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 260–273. Springer, 2013.

[New12] Mark EJ Newman. Communities, modules and large-scale structure in networks. *Nature physics*, 8(1), 2012.

[NS10] Asaf Nachmias and Asaf Shapira. Testing the expansion of a graph. *Information and Computation*, 208(4):309–314, 2010.

[NS13] Ilan Newman and Christian Sohler. Every property of hyperfinite graphs is testable. *SIAM Journal on Computing*, 42(3):1095–1112, 2013.

[OT12] Shayan Oveis Gharan and Luca Trevisan. Approximating the expansion profile and almost optimal local graph clustering. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 187–196. IEEE, 2012.

[OT14] Shayan Oveis Gharan and Luca Trevisan. Partitioning into expanders. In *SODA*, pages 1256–1266, 2014.

[OZ14] Lorenzo Orecchia and Zeyuan Allen Zhu. Flow-based algorithms for local graph clustering. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1267–1286, 2014.

[POM09] Mason A Porter, Jukka-Pekka Onnela, and Peter J Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.

[PSZ17] Richard Peng, He Sun, and Luca Zanetti. Partitioning well-clustered graphs: Spectral clustering works! *SIAM Journal on Computing*, 46(2):710–743, 2017.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[RTVX11] Ronitt Rubinfeld, Gil Tamir, Shai Vardi, and Ning Xie. Fast local computation algorithms. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 7-9, 2011. Proceedings*, pages 223–238, 2011.

[Sch07] Satu Elisa Schaeffer. Graph clustering. *Computer science review*, 1(1):27–64, 2007.

[Ses19] C. Seshadhri. Private communication. March 2019.

[SS10] Michael Saks and Comandur Seshadhri. Local monotonicity reconstruction. *SIAM Journal on Computing*, 39(7):2897–2926, 2010.

[ST13] Daniel A. Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM J. Comput.*, 42(1):1–26, 2013.

[STV99] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the xor lemma. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 537–546. ACM, 1999.

[ZLM13] Zeyuan Allen Zhu, Silvio Lattanzi, and Vahab Mirrokni. A local algorithm for finding well-connected clusters. In *Proceedings of the 30th International Conference on Machine Learning*, ICML '13, pages 396–404, 2013. Full version with title "Local Graph Clustering Beyond Cheeger's Inequality" available at `http://arxiv.org/abs/1304.8132`.

Appendix

A Further Discussions on Related Work

A.1 Relation to Testing Graph Clusterability

Both the robust clustering oracle and local reconstruction are closely related to the framework of *property testing* [RS96, GGR98]. In the bounded degree graph property testing [GR02], given a property Π, the algorithm aims to distinguish graphs that satisfy Π from graphs that are ε-far from satisfying Π by making as few queries (to the adjacency list of the graph) as possible, with high constant probability, say at least $2/3$. Here, a graph is said to be ε-far from satisfying property Π if one has to modify more than εdn edges to make it satisfy Π, while preserving the degree bound. After two decades of study, a number of properties of bounded degree graphs are now known to be testable in constant time [GR02, BSS10, HKNO09, NS13], $\tilde{O}(\sqrt{n})$ or $\tilde{O}(n^{\frac{1}{2}+c})$ time [GR98, GR00, CS10, KS11, NS10, CPS15, CKK$^+$18, KSS18].

In particular, for the property of being $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable, [CPS15] gave a testing algorithm that runs in time $\tilde{O}(\sqrt{n}\text{poly}(\phi, k, 1/\varepsilon))$ and distinguishes $(k, \phi, O(\frac{\phi^2\varepsilon^4}{k^{\Omega(1)}}))$-clusterable graphs from graphs that are ε-far from being $(k, \Theta(\frac{\phi^2\varepsilon^4}{k^{\Omega(1)}\log n}), \psi)$-clusterable, for any $\psi \in [0, 1]$. (Note that the algorithm rejects any graph that is far from clusterable graphs with *arbitrary* outer conductance.) [CKK$^+$18] recently improved this algorithm by giving an algorithm for testing if a graph contains at most k subsets with inner conductance at least ϕ from those that can be decomposed into at least $k+1$ subsets with size at least $\Omega(n/k)$ and outer conductance at most $O(\mu\phi^2)$ in time $O(n^{1/2+O(\mu)}\text{poly}(\frac{k\log n}{\phi\varepsilon}))$ for any μ that is smaller than some constant (they also generalize their algorithm for general graphs). For the case of $k = 1$, i.e., testing if the graph has expansion at least ϕ, the best known algorithm can test if a graph has expansion ϕ or is ε-far from having expansion $\Theta(\mu\phi^2)$ in time $\tilde{O}(n^{0.5+\mu})$ for any $\mu > 0$ ([KS11, NS10] which improves upon [CS10]). Furthermore, there exists a lower bound of $\Omega(\sqrt{n})$ on the query complexity for testing expansion [GR02].

Note that both the robust clustering oracle problem and the reconstruction problem are always much harder than the property testing version (see e.g., [KPS13]). For example, in the oracle problem, we need to figure out the cluster structure of the clusterable graph, and in the local reconstruction problem, the algorithm actively repairs the input graph, while the property testing is a decision problem. Furthermore, property testing only needs to distinguish between graphs which are clusterable and those are ε-far from being clusterable, while both the clustering oracle and the reconstruction have to (in some sense) approximate the distance to the class of all clusterable graphs[4]. Thus, the property testing algorithms can not be directly used to or easily modified to give a robust clustering oracle or local reconstruction algorithm. In particular, even for the case that the input graph is clusterable, one cannot use the corresponding property testing algorithm (on the clusterable graph) to answer SAMECLUSTER queries. Actually, both algorithms in [CPS15, CKK$^+$18] make decisions based on some *small summarizations* of the input graph which are constructed by a small sample of vertices and the corresponding random walk statistics. Such small summarizations can be used to distinguish if the graph is k-clusterable or is far from being k-clusterable. However, if the graph is indeed k-clusterable, they cannot be used to distinguish if two vertices are from the same cluster or are from two different clusters. As we mentioned before, in [CKK$^+$18], evidence has been provided that in general it is not possible to use pairwise Euclidean distances between two random walk distributions to distinguish between 2-clusterable graphs and far from 2-clusterable graphs if the gap between conductances is constant.

[4]Actually, in our setting, we are approximating the *intra-perturbation distance* to the class of all clusterable graphs, i.e., the minimum number of *intra-cluster* edges needed to be modified to obtain a clusterable graph over all possible h-partitions, for some $h \leq k$. This is in contrast to approximating the distance to all clusterable graphs, which is the minimum number of edges needed to be modified to obtain a clusterable graph.

On the other hand, property testing algorithms can always be obtained from the corresponding local reconstruction ones (which has already been noted in previous work on local reconstruction) and testing k-clusterability can also be obtained from our robust clustering oracle algorithm. This is also true in our scenario since we can estimate the distance between G and a clusterable graph G' with small additive error by sampling a constant number of vertices and running the oracle and clustering query algorithm (or the local reconstruction algorithm) on each sampled vertex to obtain the fraction of outlier vertices. We further note that if a graph G is ε-far from any $(k, \phi_{\text{in}}, \phi_{\text{out}})$-clusterable graph, then it cannot be an ε-perturbation of any such clusterable graph (i.e., one has to perturb more than an ε-fraction of edges). Therefore, both our robust clustering oracle and local reconstructor algorithm lead to a property testing algorithm that distinguishes $(k, \phi, O(\frac{\varepsilon\phi}{k^3 \log n}))$-clusterable graphs from graphs that are ε-far from being $(k, \Omega(\frac{\nu^k}{6^k k^4} \frac{\varepsilon\phi}{\log n}), k\nu)$-clusterable for any $\nu \in [0, 1]$, with probability at least $2/3$. The running time of the algorithm is $\tilde{O}(\sqrt{n})$, which is optimal up to polylogarithmic factors due to the $\sqrt{n}$ lower bound on the number of queries for testing expansion (corresponding to $k = 1$ in our problem) [GR02].

A.2 Other Related Work

The study on *local graph clustering* [ST13, ACL06, AP09, OT12, AOPT16, ZLM13, OZ14] is also closely related to our work. In this framework, the goal is to find a cluster from a specified vertex with running time that is bounded in terms of the size of the output set (and with a weak dependence on n). In the scenario where both inner and outer conductance are used for measuring the quality of clusters, [ZLM13] gave a local clustering algorithm that outputs a set with conductance at most $\tilde{O}(\min\{\sqrt{\phi_G(A)}, \phi_G(A)/\sqrt{\text{Conn}(A)}\})$ where A is the target set, and $\text{Conn}(A)$ is the reciprocal (e.g., $\phi(G[A])^2/(\log \text{vol}(A))$) of the mixing time of the random walk over the induced subgraph $G[A]$ on A and $\text{vol}(A)$ is the total degree of vertices in A. It is also shown that the conductance guarantee $\phi_G(A)/\sqrt{\text{Conn}(A)}$ is *tight* among (some class of) random-walk based local algorithms [ZLM13]. It might be interesting to note the logarithmic factor (i.e., $\log(\text{vol}(A))$) dependency appeared in these guarantees. The performance guarantee has later been improved by [OZ14] using a flow-based local improvement algorithm that finds a set with conductance $\psi = O(\phi_G(A))$, volume $O(\text{vol}(A))$ and runs in time $\tilde{O}(\text{vol}(A)/\psi)$, where A is the target set with $\text{Conn}(A)/\phi_G(A) = \Omega(1)$. Note that the running times of these algorithms are sublinear only if the size (or volume) of the target set is small (say, at most $o(n)$), while in our setting, the clusters of interest have at least linear size (for any constant ε).

Fully or partially recovering the clusters in the noisy model has been extensively studied in the "global algorithm regimes". Examples include recovering the planted partition in *stochastic block model* with modeling errors or noise (e.g., [CL15, GV16, MPW16, MMV16]), *correlation clustering* on different ground-truth graphs in the *semi-random* model (e.g., [MS10, CJSX14, GRSY14, MMV15]) and partitioning the graph in the *average-case* model [MMV12, MMV14, MMV15]. All these algorithms run in at least linear time.

Local reconstruction of some other properties have been investigated before. Such properties include expanders [KPS13], graph connectivity and diameter [CGR13], bipartite and ρ-clique dense graphs [Bra08], geometric properties [CS11], monotone functions [ACCL08, SS10], Lipschitz functions [JR13] and low rank matrices and subspaces [DGK17]. This algorithmic framework is also closely related to local decodable codes (e.g., [STV99]) and local decompression [DLRR13]. The local reconstruction model has been generalized to *local computation* model by Rubinfeld et al. [RTVX11, ARVX12], and a number of problems like maximal independent set, hypergraph coloring and maximum matching have been investigated in this model [RTVX11, ARVX12, MRVX12, MV13].

B Deferred Parts from Section 2

B.1 A Simple Reduction from d-Bounded Graphs to d-Regular Graphs

Given a graph G with maximum degree upper bounded by d, it will be very convenient to consider the d-regular graph G' that is obtained by adding an appropriate number of self-loops (each with half weight) to each vertex so that every vertex has degree exactly d. Note that the (normal) random walk on G we defined above is exactly the *lazy* random walk of the graph G'. Let $\mathbf{A}$ denote the adjacency matrix of G', and let $\mathbf{L} := \mathbf{I} - \frac{1}{d}\mathbf{A}$ denote the normalized Laplacian matrix of G'. We let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq 2$ denote the eigenvalues of $\mathbf{L}$ and let $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n$ denote the corresponding orthonormal (row) eigenvectors. That is, $\mathbf{L} = \sum_i \lambda_i \cdot \mathbf{v}^T \cdot \mathbf{v}$. Note that the lazy random walk matrix corresponding to G' is $\mathbf{P} := \frac{\mathbf{I}+\frac{1}{d}\mathbf{A}}{2} = \mathbf{I} - \frac{\mathbf{L}}{2}$. This implies that the eigenvalues of $\mathbf{P}$ are $1 = 1 - \frac{\lambda_1}{2}, 1 - \frac{\lambda_2}{2}, \cdots, 1 - \frac{\lambda_n}{2} \geq 0$, with corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n$. In particular, $\mathbf{P} = \sum_i (1 - \frac{\lambda_i}{2}) \cdot \mathbf{v}^T \cdot \mathbf{v}$. Furthermore, it holds that $\mathbf{p}_v^t = \mathbf{1}_v \cdot \mathbf{P}^t = \sum_i (1 - \frac{\lambda_i}{2})^t \cdot \mathbf{v}^T \cdot \mathbf{v}$.

B.2 Description of the Algorithm EstimateRCP

In the algorithm, C is a sufficiently large constant.

EstimateRCP$(G, u, v, \theta, \delta, t)$

1. Run the following $C \log n$ times:
   (a) Let $F_u :=$ FindSet(G, u, θ, t) and $F_v :=$ FindSet(G, v, θ, t)
   (b) Keep performing uniform average walks of length t from u (resp. v) until $x := \sqrt{n}/\delta^2$ such walks end at vertices in F_u (resp. F_v). Let W_u (resp. W_v) denote the set of walks. If more than $20x$ walks are performed (from either u or v), then report Fail.
   (c) Let A be the number of pairwise collisions[5] between walks in W_u and W_v. Output A/x^2.
2. If the majority of the above runs do not fail, then output the median of all the output numbers in successful runs. Otherwise, Abort.

FindSet(G, u, θ, t)

1. Perform $C\sqrt{n} \log n$ independent uniform average walks of length t from u.
2. Let F_u denote the set of all vertices w such that at most $C(1 - \frac{\theta}{2}) \log n$ walks from u end at w. Return F_u.

C Deferred Proofs from Section 3

In the following, we prove Lemma 3.2.

Proof of Lemma 3.2. First, note that if there are more than $\sqrt{\kappa}|C|$ vertices in C satisfying that $\mathbf{a}_u^t(v) \leq (1 - \sqrt{\kappa})/|C|$, then $\|\mathbf{a}_u^t - \mathcal{U}_C\|_{TV} > \sqrt{\kappa}|C| \cdot \sqrt{\kappa}/|C| \geq \kappa$, which contradicts to the fact that u is strong with respect to C.

Second, by the definition of the set $S_u^{\theta_0}$ and the fact that $\theta_0 \leq 1/2$, there can be at most $2\sqrt{n}$ vertices in $V \setminus S_u^{\theta_0}$, and thus there are at least $(1 - \sqrt{\kappa})|C| - 2\sqrt{n}$ vertices $w \in S_u^{\theta_0} \cap C$ such that $\mathbf{a}_u^t(w) \geq \frac{1-\sqrt{\kappa}}{|C|}$. Thus

$$\mathbf{a}_u^t(S_u^{\theta_0}) \geq ((1 - \sqrt{\kappa})|C| - 2\sqrt{n}) \cdot \frac{1 - \sqrt{\kappa}}{|C|} \geq 1/2,$$

[5] If a walk from W_u and a walk from W_v end at the same vertex, then this counts as one pairwise collision.

where in the second inequality we used the fact that $|C| \geq 3\sqrt{\varepsilon'}n = 3\sqrt{\frac{6\varepsilon}{\phi}}n > \frac{8\sqrt{n}}{\sqrt{\kappa}}$ as $\varepsilon = \Omega(\frac{\phi}{n})$.

Finally, since u is strong with respect to C, there are at least $(1-\sqrt{\kappa})|C| - 2\sqrt{n}$ vertices $w \in S_u^{\theta_0} \cap C$ such that $\mathbf{a}_u^t(w) \geq \frac{1-\sqrt{\kappa}}{|C|}$. The same is true for v. Thus, there are at least $(1-2\sqrt{\kappa})|C| - 4\sqrt{n}$ vertices $w \in S_u^{\theta_0} \cap S_v^{\theta_0} \cap C$ such that $p_u(w), p_v(w) \geq \frac{1-\sqrt{\kappa}}{|C|}$. Again, by the fact that $|C| > \frac{8\sqrt{n}}{\sqrt{\kappa}}$, we have that

$$\mathrm{rcp}_{\theta_0}(u,v) \quad \geq \quad ((1-2\sqrt{\kappa})|C| - 4\sqrt{n}) \cdot \frac{1-\sqrt{\kappa}}{|C|} \cdot \frac{1-\sqrt{\kappa}}{|C|} \geq \frac{1-5\sqrt{\kappa}}{|C|}.$$

This finishes the proof of the Lemma. ∎

D Further Guarantees on the Locally Reconstructed Graph

In the following, we show that by sacrificing the inner conductance quality, we can also find a clustering of the reconstructed graph G' with small outer conductance.

Lemma D.1. *Let $\phi^* = \frac{a_{4.3}\varepsilon\phi}{k^4 \log n}$. If G is an ε-perturbation of a $(k, \phi, \frac{a_{1.5}\varepsilon\kappa^4\phi}{3k^3\log n})$-clusterable graph, then the resulting graph G' from the local reconstruction algorithm is $(k, \frac{\nu^6}{6^k}\phi^*, \min\{k\nu, 1\})$-clusterable, for any $0 \leq \nu \leq 1$.*

Proof. We start with the $(k, \phi^*, 1)$-clustering of G' that is guaranteed from Lemma 4.3. Let $C_1, \cdots, C_h$ be a partition satisfying that $\phi(G'[C_i]) \geq \phi^*$. Let $\nu \in [0,1]$. We next carefully merge some of these clusters so that each part of the final partition will have both inner conductance at least $\frac{\nu^k}{6^k}\phi^*$ and outer conductance at most $\min\{k\nu, 1\}$.

If there exists $1 \leq i \neq j \leq h$ such that $|C_i| \leq |C_j|$ with $|E'(C_i, C_j)| \geq \nu d|C_i|$, then we merge C_i and C_j to obtain a new cluster $C := C_i \cup C_j$. We repeat until the condition is violated.

Note that this process always terminates as each time the number of clusters decrease by 1. Furthermore, note that after termination, each cluster has outer conductance at most $\min\{1, k\nu\}$ by construction. Now we show that in each iteration, the merged $C = C_i \cup C_j$ still has large inner conductance. Let $S \subset C$ with $|S| \leq \frac{|C|}{2}$. Let $S_i = S \cup C_i$ and $S_j = S \cup C_j$. Note that it can not happen simultaneously that $|S_i| > \frac{|C_i|}{2}$ and $|S_j| > \frac{|C_j|}{2}$. Now we have the following cases.

- If both $|S_i| \leq \frac{|C_i|}{2}$ and $|S_j| \leq \frac{|C_j|}{2}$, then

$$\phi_{G[C]}(S) = \frac{|E'(S, C \setminus S)|}{d|S|} \geq \min\{\frac{|E'(S_i, C_i \setminus S_i)|}{d|S_i|}, \frac{|E'(S_j, C_j \setminus S_j)|}{d|S_j|}\} \geq \phi^*.$$

- If $|S_j| > \frac{|C_j|}{2}$, then $|S| \leq |C_i| + |S_j| \leq |C_j| + |S_j| < 3|S_j|$.
  1. If $|S_j| \geq (1 - \frac{\nu}{2})|C_j|$, then $|C_i| \geq \frac{2}{3}|C_j|$ as otherwise $|C| \leq \frac{5}{3}|C_j|$ and $|S| \geq |S_j| > \frac{|C|}{2}$, a contradiction. Then $|S_i| \leq \frac{\nu}{2}|C_j| \leq \frac{\nu}{2}\frac{3}{2}|C_i| = \frac{3\nu}{4}|C_i|$. Thus there will be at least $\frac{d\nu}{4}|C_i|$ edges between S_j and $C_i \setminus S_i$. Thus $\phi_{G[C]}(S) \geq \frac{|E'(S_j, C_i)|}{d|S|} \geq \frac{\frac{d\nu}{4}|C_i|}{3d|S_j|} \geq \frac{\frac{d\nu}{4}\frac{2}{3}|C_j|}{3d|C_j|} = \frac{\nu}{18}$.
  2. If $|S_j| \leq (1 - \frac{\nu}{2})|C_j|$, then $|C_j \setminus S_j| \geq \frac{\nu}{2}|C_j| \geq \frac{\nu}{2(1-\frac{\nu}{2})}|S_j|$. Therefore, $\phi_{G[C]}(S) \geq \frac{|E'(S_j, C_j \setminus S_j)|}{d|S|} \geq \frac{\phi^* d|C_j \setminus S_j|}{3d|S_j|} > \frac{\phi^*\nu}{6}$.

- If $|S_i| > \frac{|C_i|}{2}$, then it must hold that $|S_j| < \frac{|C_j|}{2}$.
  1. If $|S_i| < (1 - \frac{\nu}{2})|C_i|$, then $\frac{|C_i|}{2} \geq |C_i \setminus S_i| \geq \frac{\nu}{2}|C_i|$. Thus $\phi_{G[C]}(S) \geq \frac{|E'(S_i, C_i \setminus S_i)| + |E'(S_j, C_j \setminus S_j)|}{d(|S_i| + |S_j|)} \geq \min\{\frac{\phi^* d|C_i \setminus S_i|}{d|S_i|}, \frac{\phi^* d|S_j|}{d|S_j|}\} = \min\{\frac{\nu\phi^*}{2}, \phi^*\} = \frac{\nu\phi^*}{2}$.

2. If $|S_i| \geq (1-\frac{\nu}{2})|C_i|$, then $|E'(S_i, C_j)| \geq \frac{d\nu}{2}|C_i|$. If $|E'(S_i, S_j)| \geq \frac{1}{2}|E'(S_i, C_j)|$, then $|S_j| \geq \frac{\nu}{4}|C_i|$, then $\phi_{G[C]}(S) \geq \frac{|E'(S_j, C_j \setminus S_j)|}{d|S|} \geq \frac{\phi^* d|S_j|}{d(|S_j|+|C_i|)} \geq \frac{\phi^* \nu}{5}$. Otherwise, $|E'(S_i, S_j)| < \frac{1}{2}|E'(S_i, C_j)|$, then $|E'(S_i, C_j \setminus S_j)| \geq \frac{1}{2}|E'(S_i, C_j)| \geq \frac{d\nu}{4}|C_i|$. Thus $\phi_{G[C]}(S) \geq \frac{|E'(S_i, C_j \setminus S_j)|+|E'(S_j, C_j \setminus S_j)|}{d(|S_i|+|S_j|)} \geq \min\{\frac{\frac{d\nu}{4}|C_i|}{d|S_i|}, \frac{\phi^* d|S_j|}{d|S_j|}\} \geq \min\{\frac{\nu}{4}, \phi^*\}$.

From the above analysis, we know that if both $\phi(G[C_i]) \geq \phi^*$ and $\phi(G[C_j]) \geq \phi^*$, then after merging C_i and C_j, the resulting cluster C has inner conductance at least $\frac{\nu\phi^*}{6}$. Since there will be at most k iterations (or merges), we know that in the final partition $\mathcal{P}'$, each part has outer conductance at most $\min\{k\nu, 1\}$ and inner conductance $\frac{\nu^k \phi^*}{6^k} = \frac{a_{4,3}\nu^k}{6^k k^4} \frac{\varepsilon\phi}{\log n}$. This proves the statement of the lemma. ∎